

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai proses pengumpulan data dan pengolahan data dalam penelitian serta langkah pengembangan *prototype*. Pengumpulan data pada penelitian ini lebih mengarah pada pengumpulan data yang didapatkan dari hasil wawancara dan dokumentasi pada proses *mixing* PT. Flamboyan Jaya. Pengolahan data yang dilakukan meliputi data HOQ perancangan *conveyor*, pengumpulan data alternatif serta pembuatan *prototype* aplikasi pengembangan produk menggunakan Microsoft Excel.

4.1 Gambaran Umum Perusahaan

Pada bagian ini akan dijelaskan menjelaskan tentang sedikit sejarah perusahaan, bidang usaha dan struktur organisasi pada perusahaan.

4.1.1 Profil Perusahaan

PT. Flamboyan Jaya didirikan pada tahun 2006 di daerah Cokroaminoto, Ngenep, Karangploso. Pada awal berdirinya PT. Flamboyan Jaya hanya memiliki mesin *mixing*, *blowing* dan *cutting* masing-masing berjumlah 1 buah dengan jumlah karyawan sebanyak 5 orang. Tahun 2009 PT. Flamboyan Jaya berpindah tempat di Jl. Flamboyan Raya I B 6 Songgokerto yang sekarang menjadi kantor saat ini. Jumlah mesin *cutting* dan *blowing* pada tahun 2009 bertambah menjadi 2 buah. Pada tahun 2011 PT. Flamboyan Jaya kembali berpindah tempat di Jl. Raya Donowarih RT.17/05, Karangploso, Malang hingga saat ini. Jumlah mesin *blowing* dan *cutting* saat ini sudah bertambah mejadi 5 dan 4 buah, serta jumlah karyawan saat ini berjumlah 50 orang. Karyawan-karyawan yang ada di PT. Flamboyan Jaya mayoritas berasal dari sekitar lokasi pabrik didirikan.

Nama PT. Flamboyan Jaya diambil dari nama jalan Flamboyan Raya yang menjadi lokasi tempat tinggal pendiri perusahaan dan juga sama dengan nama bidang usaha yang dimiliki oleh pendiri perusahaan. Wilayah pemasaran yang menjadi pasar utama perusahaan sejak awal berdirinya perusahaan ini berada didaerah Jombang dan nganjuk, hingga saat inirutin melakukan pengiriman ke daerah tersebut tiap hari. Sedangkan untuk *supplier* bahan baku berada didaerah Pasuruan dan Malang.

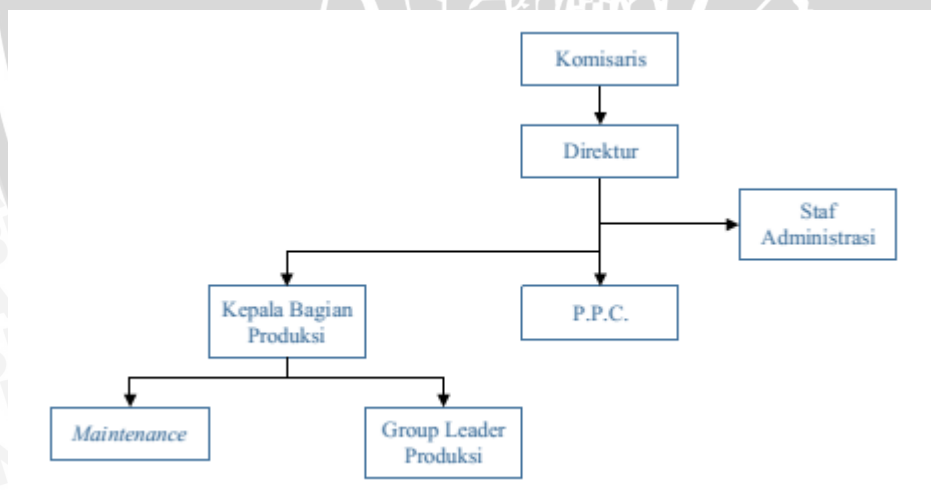
4.1.2 Bidang Usaha

PT. Flamboyan Jaya sejak awal fokus dalam memproduksi produk kantong plastik. Produk kantong plastik yang dibuat yaitu menggunakan bahan baku dari biji plastik daur ulang, sehingga kualitas yang dihasilkan tergolong kualitas KW (kualitas nomor 2). Produk yang dibuat yaitu antara lain kantong plastik jenis natural (putih), natural kuning, dan warna (biru, merah, hijau) dengan ukuran yang bervariasi 15cm, 17cm, 19cm, 21cm, dan 24cm. Rata-rata produksi paling banyak yaitu pada produk kantong plastik natural 29 putih sekitar 80% dari total produksi. PT. Flamboyan Jaya termasuk perusahaan MTO (*make to order*) yaitu memproduksi sesuai dengan permintaan konsumen.

Sejak tahun 2011 setelah pindah lokasi perusahaan, PT. Flamboyan Jaya juga memproduksi tali rafia. Awalnya PT. Flamboyan Jaya memproduksi sendiri tali rafia, namun saat ini sudah tidak memproduksi sendiri lagi dan hanya melakukan penggulungan dan *packing* tali rafia.

4.1.3 Struktur Organisasi

Pada awal didirikannya PT. Flamboyan Jaya menggunakan manajemen secara kekeluargaan, seiring bertambah besarnya perusahaan maka struktur organisasi pun di ubah menjadi lebih kompleks seperti pada Gambar 4.1 dibawah ini.



Gambar 4.1 Struktur Organisasi PT. Flamboyan Jaya

4.1.4 Proses Produksi

Proses produksi produk plastik pada PT. Flamboyan Jaya terdiri dari beberapa tahapan proses pengolahan dan juga terdapat beberapa alat-alat serta mesin yang digunakan untuk melakukan proses produksi.

4.1.4.1 Peralatan Yang Digunakan

Pada proses pembuatan kentong plastik digunakan beberapa peralatan termasuk mesin dan juga alat bantu tambahan yang menunjang sistem produksi, peralatan yang digunakan antara lain:

1. Timbangan

Timbangan digunakan sebagai alat bantu untuk mengukur berat dari produk plastik pada saat WIP (*work in process*) setelah proses *blowing* dan *cutting*, serta juga saat akan dilakukan pengemasan.

2. Meteran

Meteran berfungsi sebagai alat bantu untuk mengukur apakah dimensi produk sudah sesuai. Selain itu juga meteran digunakan bersamaan dengan timbangan untuk membandingkan panjang dan berat sudah sesuai standar yang ditentukan atau belum.

3. Mesin *Mixing*

Mesin *mixing* digunakan untuk mengaduk atau mencampur beberapa jenis bahan baku agar menjadi rata saat akan diblowng.

4. Mesin *Blowing*

Mesin *blowing* digunakan untuk mengubah bahan baku biji plastik yang berbentuk pelet menjadi cair yang nantinya dari cair dijadikan lagi berbentuk lembaran panjang/film.

5. Mesin *Cutting*

Mesin *cutting* digunakan untuk menyambungkan dua lembaran plastik hasil dari *blowing* dan juga untuk memotong lembaran sesuai ukuran yang telah ditentukan.

6. Mesin Plong

Mesin plong adalah mesin yang digunakan untuk membuang sebagian dari sisi plastik yang tujuannya untuk melubangi plastik dan membentuk pegangan kantong plastik.

7. Gunting

Gunting digunakan sebagai alat bantu untuk memotong tali rafia dan lainnya saat proses pengemasan.

4.1.4.2 Proses Pengolahan Produk

Proses pengolahan pada pembuatan produk plastik terdapat beberapa tahapan proses yaitu *mixing*, *blowing*, *cutting*, plong dan *packing*.

1. *Mixing*

Proses pengolahan pertama yaitu proses *mixing* dimana beberapa bahan baku menjadi satu dan merata. Proses ini berjalan sekitar 15 menit dan rata-rata dilakukan tiap *shift* selama 3 kali dengan sekali proses mampu mengaduk/mencampur bahan baku sebanyak 300-350 Kg.

2. *Blowing*

Pada proses *blowing* bahan baku yang telah dicampur akan dipanaskan dengan suhu sekitar 80°C - 125°C untuk menghilangkan air yang terkandung dalam bahan baku, selain itu juga mengubah bentuk bahan baku yang tadinya berbentuk pellet menjadi semi-padat yang nantinya akan dilakukan proses peniupan sehingga bahan baku cair tadi menjadi berbentuk lembaran atau film. Pada proses *blowing* ini juga terdapat produk cacat seperti mengalami gembos, kurang matang, adanya mata ikan, warna yang terlalu gelap, juga lipatan yang tidak sama dan lainnya.



Gambar 4.2 Mesin *Blowing*



Gambar 4.3 Roll Hasil Proses *Blowing*

3. *Cutting*

Proses *cutting* dimulai dari proses pengelasan atau penyambungan plastik sehingga plastik yang berupa lembaran tadi menjadi seperti kantong, setelah itu baru dilakukan pemotongan sesuai ukuran. Pada proses *cutting* ini juga operator melakukan pengecekan terhadap kualitas plastik dan plastik yang tidak memenuhi spesifikasi seperti ukuran tidak pas, mudah robek saat ditarik, dan lainnya menjadi limbah.



Gambar 4.4 Proses *Cutting*

4. *Plong*

Plong merupakan proses pembolongan kantong plastik dimana plastik yang sebelumnya disatukan berbentuk persegi akan dipotong ujungnya agar terbentuk pegangan kantong plastik. Pada proses ini terdapat limbah yang paling banyak karena sisa pembolongan ini tidak dapat dijadikan kantong plastik atau digunakan lagi dalam proses. Sisa pembolongan ini sekitar 10% dari luas kantong plastik.



Gambar 4.5 Proses *Plong*

5. *Packing*

Proses pengemasan (*packing*) dimulai dari penimbangan produk yang telah jadi, lalu selanjutnya dilakukan pengikatan dengan tali rafia sesuai dengan berat yang telah

ditentukan, setelah diikat lalu dimasukan kedalam karung plastik sesuai dengan pesanan.



Gambar 4.6 Proses Packing

4.2 Pengumpulan Data

Data dihimpun dari penelitian yang dilakukan oleh Darmawan (2015) “perancangan *flexible roller conveyor* sebagai *material handling device* yang ramah energi” pada kota malang ialah sebagai berikut :

4.2.1 Technical Importance

Berikut ialah *Technical Response* yang diperoleh dari data penelitian oleh Darmawan (2015) seperti terlihat pada Tabel 4.1.

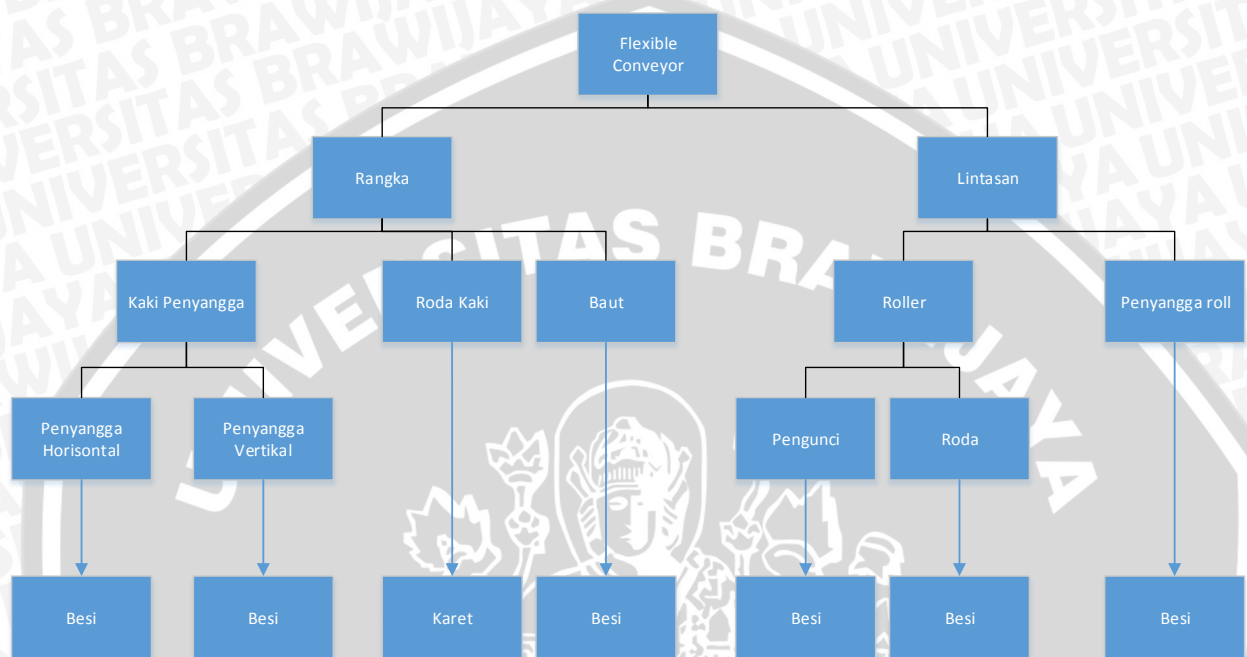
Tabel 4.1 *Technical Importance*

No	Technical response	Value	Unit
1	Desain konveyor dibuat fleksibel tidak permanen	Sliding Lock, Bongkar Pasang	Lipat,
2	Konveyor memiliki Ketinggian yang dapat diatur sesuai dengan pengguna	150-100	Cm
3	Bahan konveyor ringan dan kuat	Tungsten, Besi Eser, Aluminum	N/m ²
4	Konveyor mudah dipindahkan.	360 Caster Wheel, Rubber Wheel, Caster Wheel	
5	Desin konveyor ramah lingkungan dan tidak mengeluarkan energi buangan saat dioperasikan.	Gravitasi, Thermal, Dinamo Listrik	Joule

Dari data diatas dapat diketahui ada 5 respon teknis yang dimiliki oleh *flexible conveyor*. Dari desain konveyor yang bersifat flexible, memiliki bahan yang bersifat kuat dan ringan, dengan ketinggian yang dapat diatur, sekitar 150-100 cm. Dengan desain konveyor ramah lingkungan dengan pilihan Gravitasi, *thermal*, dinamo listrik.

4.2.2 BOM Tree

Berdasarkan data yang diperoleh dari Penelitian tersebut ialah sebagai berikut



Gambar 4.7 BOM Tree Flexible Conveyor

Dari Gambar 4.7 diketahui bahwa *Flexible Conveyor* memiliki 2 komponen utama, yaitu rangka dan lintasan. Komponen rangka memiliki sub *part* kaki penyangga, roda kaki, dan baut. Sementara Lintasan memiliki sub *part* roller dan penyangga roll. Sub *part* kaki penyangga masih memiliki *breakdown part*, yaitu penyangga horizontal dan vertikal, sementara *roller* memiliki subpart pengunci dan roda.

4.3 Tahapan Perancangan Sistem

Langkah-langkah perancangan aplikasi pengembangan produk adalah sebagai berikut:

1. Menetapkan tujuan *prototype*

Penetapan tujuan dan manfaat dari *prototype* telah dilakukan pada bab I, yakni mencari pembangkitan jumlah kemungkinan alternatif kombinasi desain dari *morphology chart* dan pembuatan Diagram FAST.

2. Mendefinisikan fungsi *prototype*

- a. *Requirement modelling*

Sistem Requirement Checklist yang dibuat nantinya mewakili *user requirement* SPV Bagian Produksi PT Flamboyan Jaya.

b. *Data modelling*

Pada langkah ini, aplikasi pengembangan produk digambarkan dengan aliran data dan informasi yang dihasilkan dari sistem dengan DFD agar nampak jelas.

c. *Process modelling*

Menggambarkan logika atau aturan bisnis dalam aplikasi pengembangan produk pada PT Flamboyan Jaya

d. *Development Strategies*

Dalam langkah ini, akan ditentukan kebutuhan minimum *software* dan *hardware*, level pembuatan sistem, kebutuhan minimal sistem operasi dan kebutuhan minimal administrator.

3. Mengembangkan *prototype*

a. Langkah Desain

1) Desain *Database Logis*

Dalam tahap ini akan dibuat *list entity* yang telah dirancang pada tahap *data modelling*.

2) Desain *Database Fisik*

Desain fisik merupakan aktualisasi dari desain logis. Disini *table* disesuaikan dengan rancangan bentuk fisik yang sesuai dengan Aplikasi yang digunakan.

3) Desain *User Interface (UI)*

Desain UI merupakan desain kerangka awal *display* tatap muka pengguna aplikasi pengembangan produk. Tahapan ini meliputi Desain menu *windows*, alur *user interface*, navigasi *user interface*, kebutuhan tombol, kebutuhan *form*, *visual output* dan kerangka awal tiap *windows*.

4) Desain Algoritma

Algoritma yang ditampilkan merupakan algoritma yang digunakan untuk membuat Diagram FAST dan *Morphology Chart*

b. Implementasi, membuat aplikasi yang dirancang dengan melakukan pengembangan *database*, *module*, dan *user interface* menggunakan VBNET.

4. Mengevaluasi *prototype*

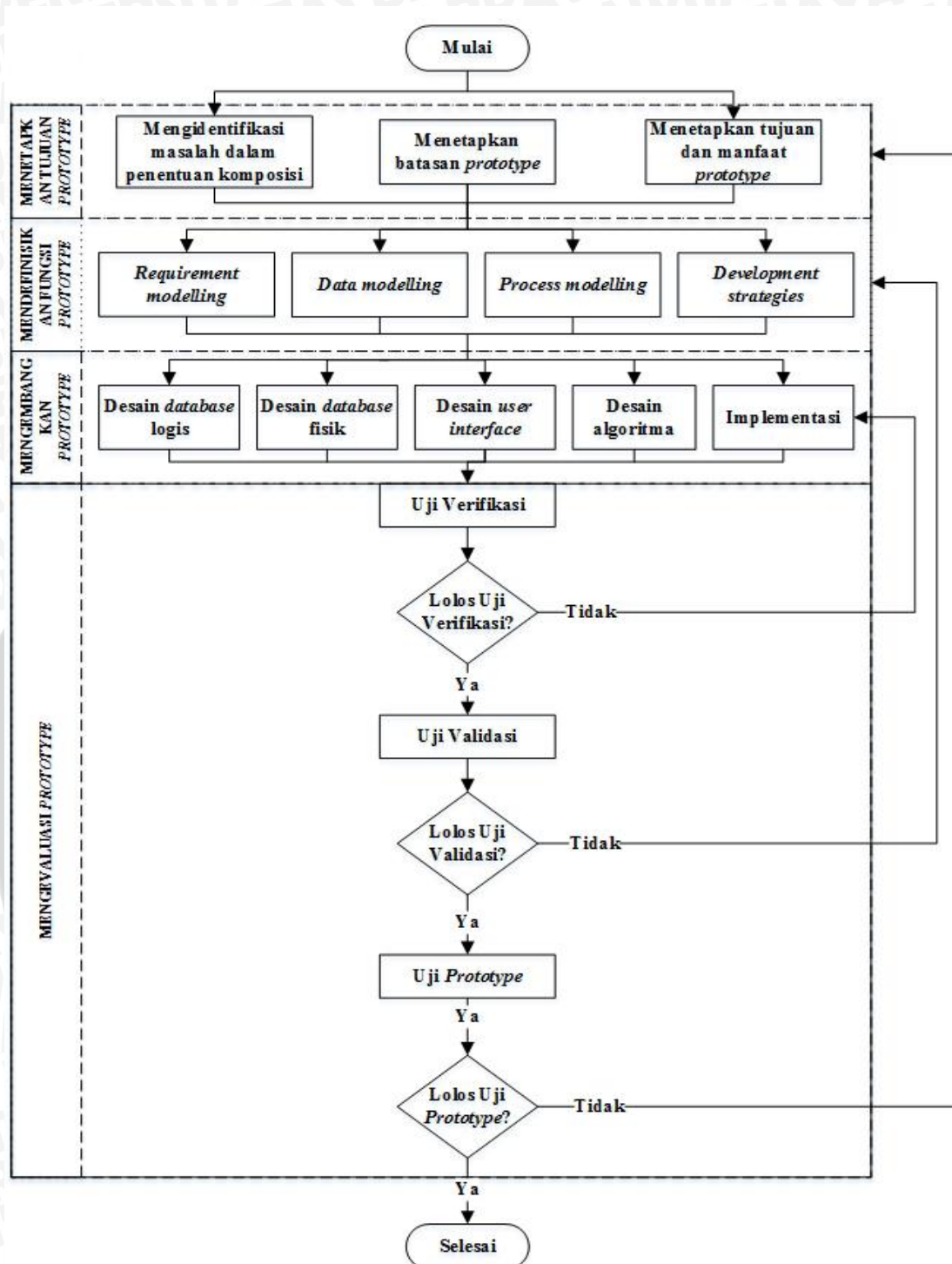
Pada langkah ini dilakukan pengujian terhadap program aplikasi yang telah dibuat. Pengujian program ini ditinjau dari tiga segi, yaitu verifikasi, validasi dan uji *prototype*.

- a. Verifikasi, dengan membandingkan desain *database* fisik, desain *user interface* dan desain algoritma yang dirancang dengan implementasi.
- b. Validasi, dengan membandingkan kebutuhan sistem yang telah terpenuhi dengan *System Requirement Checklist*.
- c. Uji *prototype* dengan membandingkan *prototype* sistem baru apakah dapat mengatasi kelemahan sistem lama yang dipaparkan sebagai hasil analisis PIECES.

Berikut ini merupakan diagram alir perancangan sistem yang dilakukan di PT. Flamboyan Jaya, seperti yang ditunjukkan pada Gambar 4.8.

4.4 Analisa Sistem

Penerapan sistem baru diharapkan mampu membantu tahan perancangan produk dengan melakukan pengembangan produk dengan penentuan Diagram FAST dan *Morphology Chart*, berkaca dengan sistem yang berjalan saat ini didalam setiap perancangan produk untuk tahap pengembangan produk dilakukan secara manual dan berdasarkan pengetahuan pribadi saja dan belum adanya pengetahuan dari masa lalu, hal ini menyebabkan ketidaksesuaian spesifikasi dari produk akhir yang menyebabkan banyaknya masalah yang timbul dikemudian hari. Dari analisa tersebut, penerapan aplikasi pengembangan produk di PT. Flamboyan Jaya akan menjadi penyelesaian permasalahan dari pengembangan produk yang optimal. Langkah awal dalam perancangan aplikasi ini adalah merencanakan tujuan *prototype* yang telah dijabarkan pada Bab I. Langkah selanjutnya adalah menganalisa fungsi *prototype* yang meliputi pembuatan model kebutuhan sistem (*requirement modelling*), model data (*data modelling*), dan model proses (*process modelling*).



Gambar 4.8 Diagram alir perancangan sistem
Sumber: Sommerville (2011)

4.4.1 Model Kebutuhan Sistem (*Requirement Modelling*)

Tahap ini bertujuan untuk mengidentifikasi kebutuhan PT. Flamboyan Jaya dalam pengembangan produk fase morphology chart . Model kebutuhan sistem ini dinyatakan dalam *System Requirement Checklist* (SRC) sebagai dasar untuk mengukur keberhasilan aplikasi yang akan dibangun. Model kebutuhan sistem ini digambarkan ke dalam lima

kategori umum yaitu *output*, *input*, *process*, *performance*, dan *control*. Berikut ini adalah model kebutuhan sistem yang ditunjukkan pada Tabel 4.2.

System requirement checklist untuk yang dijabarkan pada Tabel 4.1 menunjukkan bahwa kebutuhan sistem yang paling vital adalah *output* yang menyajikan informasi berupa diagram FAST, *Morphology Chart* dan Alternatif Kombinasi Desain.

4.4.2 Data and Process Modelling

Tahap ini bertujuan untuk mengembangkan model grafis untuk melihat sistem secara keseluruhan dalam mengubah data menjadi informasi yang disebut *data modelling*. Sedangkan logika proses bisnis yang terjadi digambarkan dalam *process modelling* sehingga dihasilkan model logis yang mendukung operasi bisnis yang memenuhi kebutuhan pengguna.

Tabel 4.2 System Requirement Checklist

Komponen	Penjabaran
Input	User dapat memasukkan data berikut ini: 1. Keinginan Pelanggan. 2. Fungsi yang diinginkan Pelanggan 3. Alternatif
Output	Sistem dapat memberikan <i>output</i> yang dibutuhkan, antara lain: 1. <i>Morphology Chart</i> . 2. FAST Diagram. 3. Alternatif Kombinasi Desain
Process	Sistem dapat menentukan <i>Morphology Chart</i> sesuai dengan kebutuhan agar. Dengan melihat basis data yang pernah dimasukan kedalam sistem, sehingga mempermudah pembuatan Morphology Chart Sistem dapat menunjukan Diagram FAST sesuai dengan fungsi yang dipilih oleh user hingga bagaimana cara mendapatkan fungsi tersebut.
Performance	Sistem memungkinkan dalam melakukan update data jika terdapat data baru.
Control	User diberi <i>username</i> dan <i>password</i> untuk <i>login</i> sistem yang hanya dapat diakses oleh User itu sendiri.

4.4.2.1 Data Modelling

Aplikasi pengembangan produk ini dimodelkan dengan *Data Flow Diagram* (DFD) yang bertujuan untuk menggambarkan proses bisnis yang dilakukan serta bagaimana aliran data berpindah diantara aktivitas-aktivitas tersebut. Langkah-langkah yang dilakukan dalam pembuatan DFD adalah sebagai berikut.

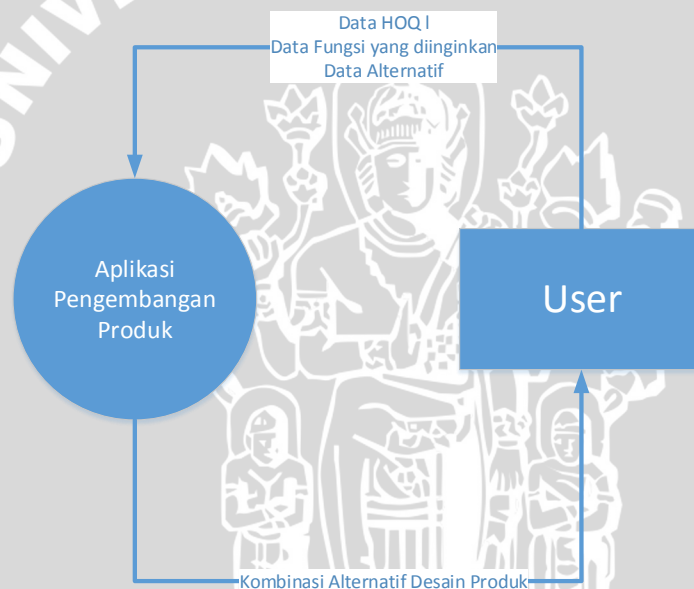
1. Identifikasi kesatuan entitas luar (*external entities*) dalam sistem, yakni *User*.
2. Langkah selanjutnya adalah identifikasi semua *input* dan *output* dari kesatuan luar yang terlibat dalam sistem seperti yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Identifikasi *Input* dan *Output* DFD

Kesatuan Luar	Input	Output
User	Data <i>Technical Response</i> dari HOQ	Kombinasi alternatif desain produk

3. Context Diagram

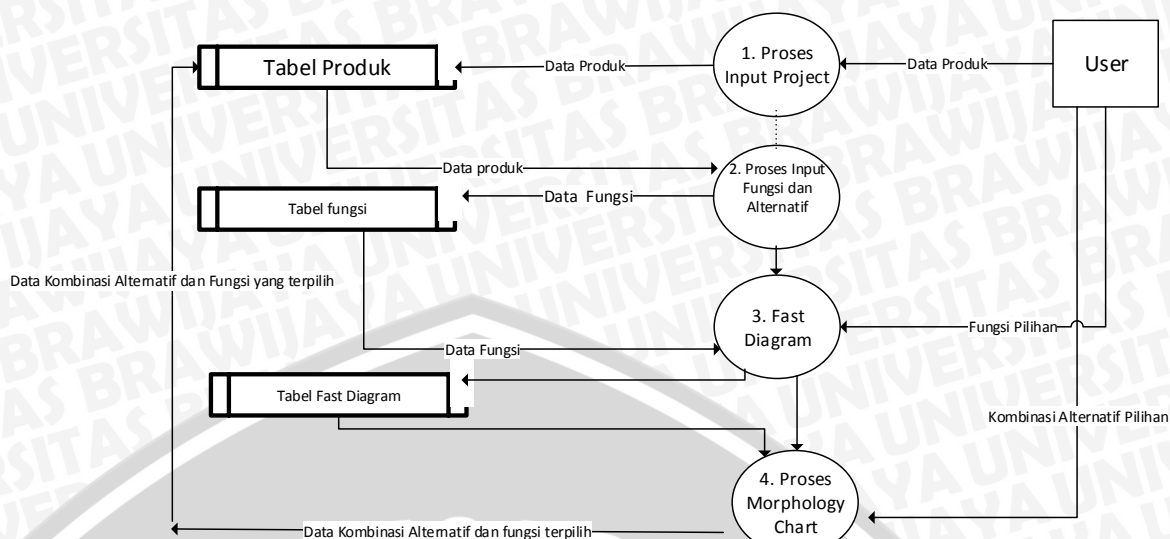
Context Diagram adalah diagram dalam rangkaian suatu DFD yang menggambarkan entitas-entitas yang berhubungan dengan suatu sistem. *Context diagram* menggambarkan hubungan antara *input* dan *output* serta *external entitites*. Gambar 4.8 berikut ini adalah *context diagram* aplikasi pengembangan produk.

Gambar 4.9 *Context Diagram* aplikasi pengembangan produk

Berdasarkan Gambar 4.9, proses yang terjadi pada *context diagram* aplikasi pengembangan produk dapat dijelaskan sebagai berikut:

- a. *User* memasukkan keinginan pelanggan , data alternatif dan data fungsi.
 - b. Sistem mengolah data tersebut, sistem kemudian menghasilkan *output* berupa diagram FAST dan *Morpholgy Chart*
- ### 4. DFD level 1

DFD level 1 menunjukkan semua proses dalam *context diagram* sekaligus bagaimana aliran informasi berpindah dari satu proses ke proses lainnya. Proses yang terjadi pada DFD level aplikasi pengembangan produk dapat dijelaskan dengan Gambar 4.9.



Gambar 4.10 DFD Level 0 aplikasi pengembangan produk

Berdasarkan Gambar 4.10 diatas proses yang terjadi pada DFD Level 0 dapat dijelaskan sebagai berikut

- Proses Input Project*
Pada proses ini user memasukan data identitas *project* seperti tanggal pembuatan *project* dan nama *project*
- Proses Input Technical Response*
Pada proses ini user memasukan data dari *technical response* yang didapat dari HOQ *project* tersebut
- Proses FAST Diagram*
Pada proses ini user akan memilih fungsi mana yang akan dimasukan kedalam proses sebelumnya.
- Proses Morphology Chart*
Pada Proses ini user akan ditunjukan semua kombinasi dari alternatif fungsi yang telah dipilih oleh user pada proses sebelumnya

4.4.2.2 Process Modelling

Process modelling merupakan langkah-langkah pengolahan dan logika yang bertujuan untuk menggambarkan proses atau aturan yang berlaku aplikasi pengembangan produk ini.

1. Sistem akan memberikan kemungkinan kombinasi alternatif dari setiap fungsi yang dipilih oleh *user*.
2. Jumlah alternatif yang dapat dimasukkan kedalam aplikasi hanya terbatas 3 alternatif.
3. Data yang dimasukkan oleh user berupa *technical response* dari proyek tersebut.

4.4.3 Development Strategies

Development strategies adalah kebutuhan untuk sistem baru yang mencakup kebutuhan *software* dan kebutuhan *hardware*. Berikut merupakan strategi pengembangan aplikasi pengembangan produk yang disesuaikan dengan kondisi pada PT. Flamboyan Jaya.

1. Level pengembangan *prototype*

Aplikasi pengembangan produk merupakan aplikasi *stand alone* yang hanya membutuhkan 1 buah komputer pada bagian produksi sehingga sistem ini dapat diakses dengan mudah.

2. *Software requirement*

Kebutuhan perangkat lunak yang dibutuhkan untuk menjalankan aplikasi ini dapat dijelaskan pada Tabel 4.4. Perangkat lunak yang dipilih tidak sulit untuk dipenuhi karena sifat dari perangkat lunak berikut ini sudah banyak digunakan dalam kegiatan sehari-hari.

Tabel 4.4 Kebutuhan Minimum Software

Unsur	Pilihan	Alasan
<i>Operating System</i>	Windows 7	1. Operating Sistem yang banyak digunakan di Indonesia, dan juga digunakan pada komputer di PT. Flamboyan Jaya.
<i>Database</i>	<i>Microsoft Access</i>	1. <i>Microsoft Access</i> sebagai database dikarenakan kemudahannya dalam mengolah proses dan umum bagi orang awam menggunakannya
Bahasa Pemrograman	<i>Visual Basic for Application (VBA)</i>	1. <i>Visual Basic for Application (VBA)</i> mudah digunakan karena sudah menyatu dengan <i>Microsoft Office</i>

3. *Hardware requirement*

Kebutuhan perangkat keras yang dibutuhkan untuk mendukung aplikasi ini dapat dijelaskan pada Tabel 4.5.

Tabel 4.5 Kebutuhan Minimum Hardware

Unsur	Pilihan	Alasan
Server	Intel Pentium Processor B925C (4M Cache, 2.00 GHz) atau lebih tinggi.	Dapat menjalankan Windows 7 dengan lancar.
	RAM 1 Ghz atau lebih tinggi.	
	Kapasitas hard disk minimal 100 Mb.	
Input	Mouse dan keyboard	Sebagai peralatan untuk memasukkan dan memproses data
Output	Monitor	Sebagai peralatan <i>display</i> atau antar muka

4. Dokumentasi

Dokumentasi yang dilakukan dalam pengembangan aplikasi pengembangan produk ini antara lain *manual book* untuk memudahkan penguasaan kompetensi dan *source code program* untuk memudahkan *upgrade* program.

5. Keterampilan User

User harus memiliki minimal kemampuan dalam mengoperasikan *Windows* dan *Microsoft Office* terutama *Microsoft Access* untuk menggunakan sistem ini karena cara pengolahan data di sistem ini tidak terlalu rumit.

Berdasarkan analisis sistem dengan menganalisa sistem lama, *requirement modelling*, *data modelling*, *process modelling*, dan *development strategies* maka spesifikasi kebutuhan sistem telah diketahui. Hasil dari analisis sistem tersebut akan dibawa ke tahap selanjutnya yaitu tahap desain dan implementasi *prototype* yang terdiri dari langkah desain, implementasi, pengujian, dan evaluasi *prototype*.

4.5 Desain Sistem

Desain sistem bertujuan untuk menciptakan model fisik dari sistem yang memenuhi persyaratan desain yang ditetapkan pada tahap analisa sistem. Tahap ini merupakan sebuah model logis dari sistem yang baru dikembangkan dari tahap analisa sebelumnya. Langkah yang akan dikerjakan pada tahap ini mencakup analisa data *input*, desain *database*, *user interface*, dan desain algoritma. Pada tahap ini akan dihasilkan spesifikasi desain dari aplikasi pengembangan produk.

4.5.1 Desain Database

Tahap desain *database* yang perlu dilakukan oleh analis secara umum adalah menganalisa terlebih dahulu *file-file* yang diperlukan oleh aplikasi. Pada umumnya

langkah-langkah untuk mendesain *database* adalah dengan desain *database* logis, normalisasi, dan desain *database* fisik.

4.5.1.1 Desain Database Logis

Fungsi – fungsi dari sistem dijelaskan kepada *user* melalui desain *database* logis. Pada tahapan ini *user* bisa mengetahui bagaimana sistem bekerja secara logika. Desain *database* logis ini digambarkan dengan menggunakan ERD, dimana ERD dapat menggambarkan entitas dan atribut yang terlibat dalam sistem. Berikut tahapan – tahapan dalam pembuatan ERD .

1. List Entity

Setiap entitas merupakan calon tabel yang merupakan komponen penyusun utama dalam sebuah *database*. Daftar entitas perlu disusun agar dapat diketahui data apa saja yang akan terlibat dalam *database*. Identifikasi terhadap entitas adalah langkah pertama yang harus dikerjakan dalam membuat daftar entitas. Berikut Tabel 4.6 adalah entitas dan atribut yang terlibat pada aplikasi perancangan produk.

Tabel 4.6 Daftar Entitas dan Atribut ERD

Entitas	Atribut
Produk	nama_produk, id_produk, nama, id_fungsi, id_user
desain Alternatif	id_kombinasi, id_alternatif, id_morphology, id_fungsi
Fungsi	Id_alternatif, id_fungsi ,id_alternatif
Alternatif	Nama_alternatif, id_alternatif
User	Id_user, Username, Password

Entitas yang terdapat dalam sistem informasi manajemen *preventive maintenance* merupakan data utama yang merupakan nama tabel dalam *database*.

2. Entity-Relational Diagram (ERD)

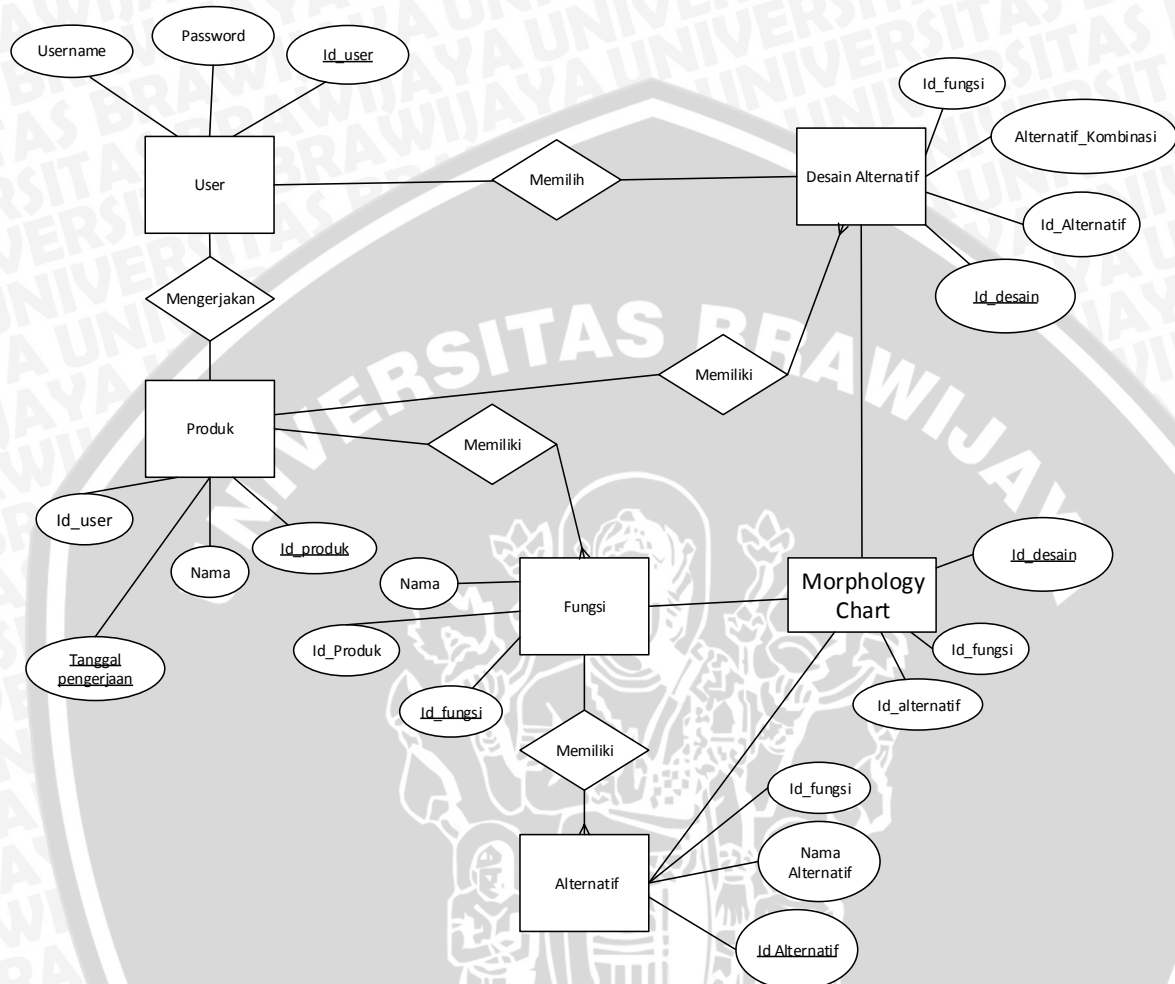
Dalam penelitian perancangan sistem ini, hubungan atau *relationship* antar objek/entitas digambarkan melalui ERD. ERD sangatlah penting, karena pada tahap ini semua entitas diintegrasikan menjadi satu sehingga dapat menghasilkan informasi yang dibutuhkan oleh *user*. Namun sebelum menggambarkan ERD, yang perlu dilakukan sebelumnya adalah mengidentifikasi kardinalitas atau jenis relasi untuk masing-masing entitas yang terlibat seperti pada Tabel 4.7.

Tabel 4.7 Identifikasi Relasi

Entitas	Relasi	Entitas	Derajat Relasi Maks -Min
User	Mengerjakan	Produk	(1 , N)
Produk	Mempunyai	Fungsi	(1 , N)
Fungsi	Mempunyai	Alternatif	(1 , N)

Produk	Memiliki	Desain Alternatif	(1, N)
--------	----------	-------------------	----------

Berdasarkan identifikasi jenis relasi yang dilakukan, kardinalitas atau jenis relasi yang terjadi terhadap masing-masing entitas dapat digambarkan pada Gambar ERD 4.11 disertai dengan atribut-atribut yang menyertai masing-masing entitas



Gambar 4.11 Diagram E-R dengan atribut

Berdasarkan Gambar 4.11 dan Tabel 4.8, hubungan antar masing-masing entitas yang terlibat dapat dijelaskan sebagai berikut:

- Produk dapat memiliki lebih dari satu kombinasi alternatif
- Produk dapat memiliki lebih dari satu fungsi
- Fungsi dapat memiliki lebih dari satu alternatif
- Fungsi dapat memiliki lebih dari satu morphology chart
- Morphology Chart dapat menghasilkan lebih dari satu kombinasi alternatif

4.5.1.2 Desain Database Fisik

Desain fisik merupakan aktualisasi dari desain database logis yang sangat bergantung dengan software yang dipakai. Oleh karena itu pada tahap ini diawali dengan pemilihan software yang akan dipakai terlebih dahulu yaitu Microsoft Access 2013. Setelah dilakukan

pemilihan maka dirancanglah struktur tabel sesuai dengan kebutuhan untuk melakukan penyimpanan data. Pada tahap ini, *entity* sudah berubah menjadi tabel sesuai dengan format tabel pada *Microsoft Access 2013*. Berikut merupakan rancangan struktur *database* dimana entitas sudah diubah menjadi tabel.

1. Entitas User

Entitas User berisi data user produk, Pada tabel ini informasi user seperti nama pengguna, *id_user*, dan password.

Tabel 4.8 Desain Database Entitas User

Field	Data Type	Field Size	Note	Key
<i>Id_user</i>	<i>Number</i>	<i>Long Integer</i>	Id pengguna	PK
<i>Username</i>	<i>Text</i>	255	Nama produk	
<i>Password</i>	<i>Text</i>	255	Sandi Pengguna	

2. Entitas Produk

Entitas Produk berisi data produk , pada tabel ini informasi produk seperti *id_user*, *id_produk*, nama produk, dan *id_kombinasi*.

Tabel 4.9 merupakan desain spesifikasi fisik untuk entitas produk.

Tabel 4.9 Desain Database Entitas Produk

Field	Data Type	Field Size	Note	Key
<i>Id_produk</i>	<i>Number</i>	<i>Long Integer</i>	Nomor Produk	PK
<i>Id_user</i>	<i>Number</i>	<i>Long Integer</i>	Id pengguna	FK
<i>Nama Produk</i>	<i>Text</i>	255	Nama produk	
<i>Id_kombinasi</i>	<i>Number</i>	<i>Long Integer</i>	Nomor Kombinasi desain	FK

3. Entitas Fungsi

Entitas *fungsi* berisi data *fungsi* produk. Pada tabel ini informasi pengguna seperti nama, id fungsi, nama fungsi dan *id_alternatif* dari user seperti yang ditunjukkan pada Tabel 4.10

Tabel 4.10 Desain Database Entitas Fungsi

Field	Data Type	Field Size	Note	Key
<i>Id_fungsi</i>	<i>AutoNumber</i>	<i>Long Integer</i>	Nomor pengguna	PK
<i>Id_alternatif</i>	<i>Text</i>	100	Nomor Alternatif	FK
<i>Nama</i>	<i>Text</i>	255	Jenis Fungsi	

4. Entitas Alternatif

Entitas mekanik merupakan data yang berisi informasi fungsi yang dimiliki oleh produk meliputi *id_alternatif*, dan Nama Alternatif. Tabel 4.10 merupakan desain spesifikasi fisik untuk entitas alternatif.

Tabel 4.10 Desain Database Entitas alternatif

Field	Data Type	Field Size	Note	Key
<i>Id_alternatif</i>	<i>Number</i>	<i>Long Integer</i>	nomor identitas alternatif	PK
<i>Nama alternatif</i>	<i>Text</i>	255	Nama dari alternatif	

5. Entitas Morphology Chart

Entitas *Morphology Chart* merupakan data yang berisi informasi *morphology chart* berupa,

Tabel 4.10 Desain Database Entitas alternatif

Field	Data Type	Field Size	Note	Key
<i>Id_alternatif</i>	<i>Number</i>	<i>Long Integer</i>	nomor identitas alternatif	Fk
<i>Id_fungsi</i>	<i>Text</i>	<i>Long Integer</i>	Nomor identitas fungsi	Fk
<i>Id_produk</i>	<i>Text</i>	<i>Long Integer</i>	Nomor identitas produk	fk

6. Entitas Kombinasi Desain

Entitas kerusakan merupakan tabel data yg dimiliki oleh kombinasi desain.

Tabel 4.12 Desain Database Kombinasi Desain.

Field	Data Type	Field Size	Note	Key
<i>Nama Kombinasi</i>	<i>Text</i>	255	Nama dari kombinasi	
<i>Id_Kombinasi</i>	<i>Number</i>	<i>Long Integer</i>	Nomor dari alternatif	FK
<i>Id_fungsi</i>	<i>Number</i>	<i>Long Integer</i>	Nomor dari fungsi	FK
<i>Id_altenratif</i>	<i>Number</i>	<i>Long Integer</i>	Nomor dari Alternatif	PK

4.5.2 Desain User Interface

Tahap ini bertujuan untuk membuat rancangan dari tampilan antarmuka yang nantinya memudahkan *user* yang berinteraksi secara langsung dengan sistem. Desain *user interface* ini dilakukan pada aplikasi *Visual Basic*. Desain *User interface* meliputi *list windows*,

hirarki *windows* dan alur *user interface*, desain navigasi antar *windows*, desain *form input*, desain *visual output* dan Desain awal *windows*. dimana harus memenuhi kebutuhan pengguna dan *user friendly* sehingga pengguna atau pemakai dapat berkomunikasi dengan sistem yang dibangun.

4.5.2.1 List Windows

List Windows adalah daftar *Windows* apa saja yang akan dirancang untuk memudahkan *user* dalam menggunakan aplikasi. Disini *Windows* digunakan sebagai dasar *user interface*. Pada *List windows* ini juga disertakan fungsi dari tiap-tiap *Windows*. Tabel 4.12 menjelaskan *Windows* apa saja yang akan digunakan pada aplikasi pengembangan produk.

4.5.2.2 Hirarki dan Alur User Interface

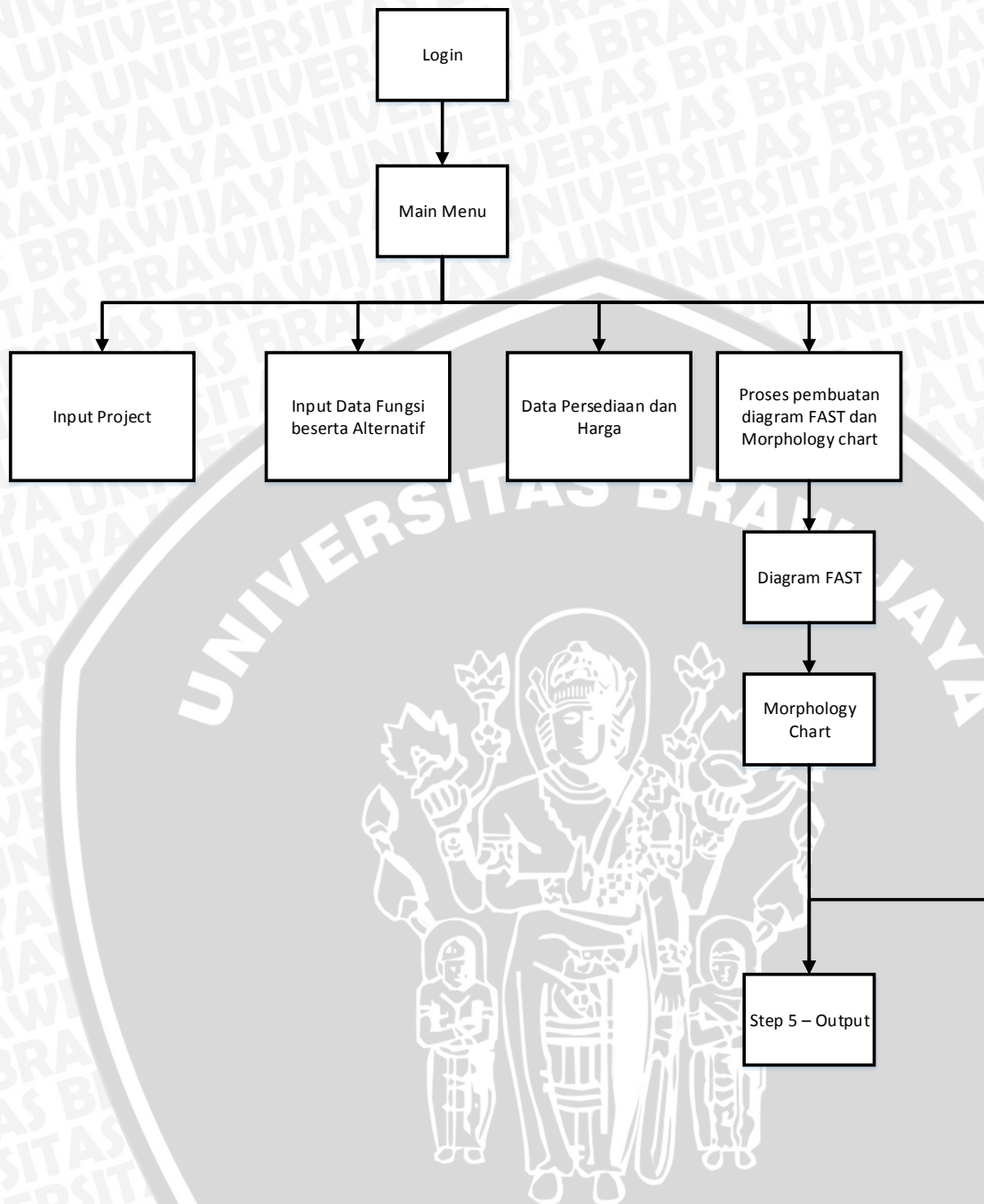
Hirarki dan alur *User Interface* menjelaskan hirarki dari tiap UI dan aliran proses dari *windows ke windows* lain yang dilihat oleh *user*. Hierarki *windows* menunjukkan *windows* mana saja yang harus ditempuh sebelum *windows* selanjutnya. Alur *user interface* menunjukkan *windows* mana saja yang akan dilihat dari proses *input* hingga keluarnya *output*, dan juga bagaimana navigasi bekerja dalam perpindahan dari tiap *windows*. Berikut merupakan hirarki *windows* yang ada pada Gambar 4.12.

Tabel 4.12 List Windows dan Fungsinya

No.	Nama Windows dan Tab	Fungsi Windows	Status Windows
1.	Login	Window ini menampilkan <i>form</i> login dimana berfungsi sebagai security check terhadap <i>user</i> .	Terlihat
2.	Main Menu	Tab ini menampilkan berbagai menu pilihan yang disajikan oleh sistem. Terdapat menu pemilihan produk yang akan diproses ataupun pembuatan <i>project</i> baru	Terlihat
3.	Input Data Project	Windows ini menampilkan <i>form input</i> data <i>project</i> . <i>Form input</i> tersebut nantinya akan mengisi data pada tabel Data <i>project</i> di <i>main menu</i> .	Terlihat
4.	Data fungsi	Tab ini menampilkan Tabel yang menunjukan fungsi fungsi yang dimiliki oleh suatu produk	Terlihat

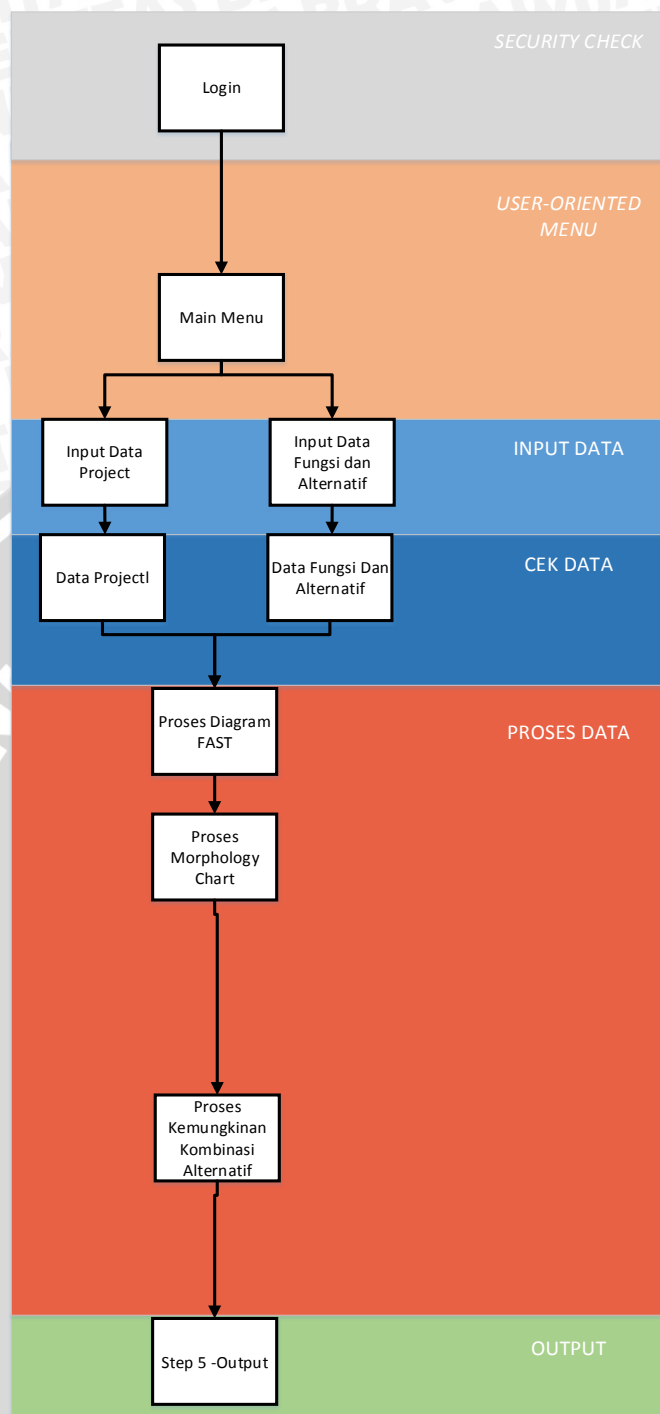
		beserta alternatifnya.	
5.	Input Data Fungsi	<i>Windows</i> ini menampilkan <i>form input data fungsi</i> . <i>Form input</i> tersebut nantinya akan mengisi data pada tabel Data project di tab data fungsi.	Terlihat
6	Tab Diagram FAST	<i>Tab</i> ini menunjukkan bagaimana data fungsi olah oleh sistem menjadi <i>diagram fast</i>	Terlihat
7	Tab <i>Morphology Chart</i>	<i>Tab</i> ini memnunjukkan <i>Morphology Chart</i> memnunjukkan bagaimana data alternatif yang diinput pada <i>windows</i> sebelumnya.	Terlihat





Gambar 4.12 Hirarki Menu *Prototype* Aplikasi pengembangan produk

Berdasarkan Gambar 4.12 diketahui bahwa terdapat 3 *tab* yang muncul setelah pengguna tiba di *windows*. Alur dari *User Interface* memuat *Step by Step* yang dilihat oleh *user* dari awal proses *input* hingga *output* keluar. Berikut merupakan alur dari *User Interface* yang dijabarkan pada Gambar 4.13.

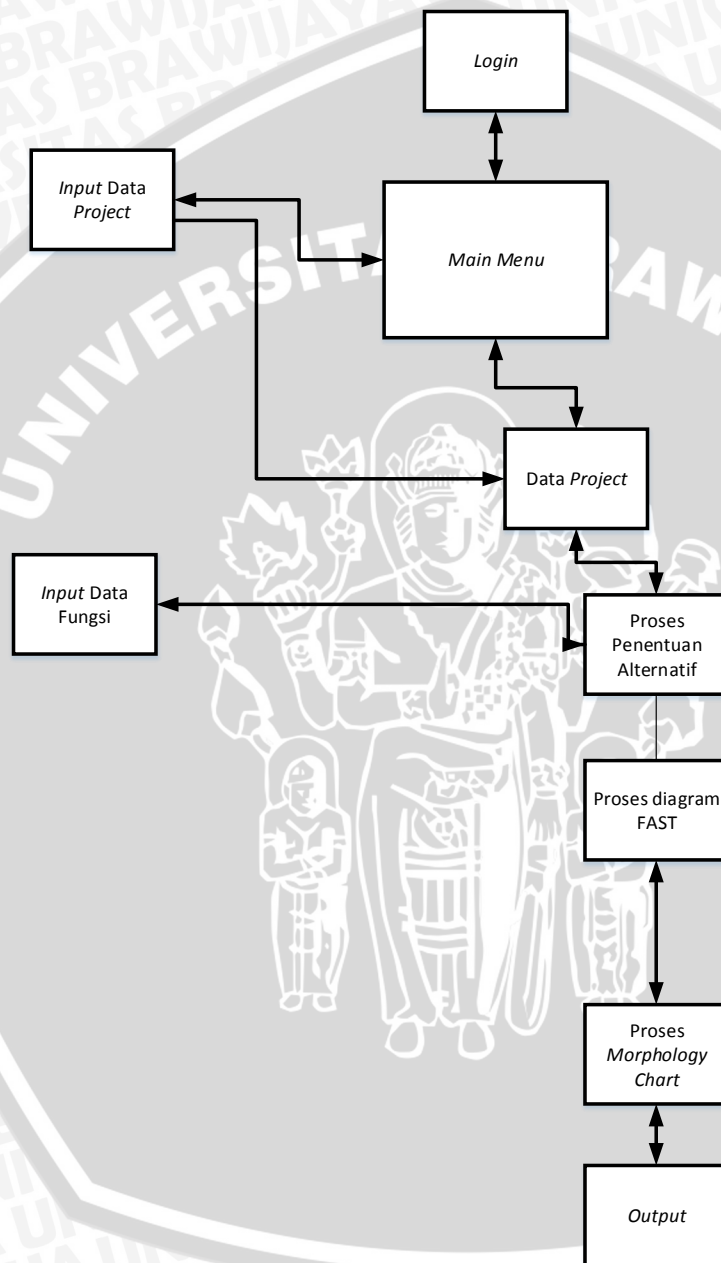


Gambar 4.13 Alur *User Interface Prototype* Aplikasi Pengembangan Produk

Berdasarkan Gambar 4.13 dapat dilihat bahwa dalam terdapat ada beberapa tahap dari alur *User Interface Prototype* Aplikasi pengembangan produk. User akan bertemu dengan *main menu*, lalu akan dilanjutkan input data, pada *Input* data project dan *input* data fungsi dan alternatif proses cek data, dilanjutkan proses data pada proses diagram fast dan *morphology chart* hingga *Output*.

4.5.2.3 Desain Navigasi antar Menu

Desain navigasi antar menu bertujuan untuk memberikan *control* dan kemudahan kepada *user* dalam menggunakan fungsi aplikasi secara maksimal. Pembuatan desain navigasi ini berdasarkan alur *user interface* dan hirarki *tab*. Berikut dijabarkan desain navigasi antar menu pada Gambar 4.14.



Gambar 4.14 Desain Navigasi Antar Menu *Prototype* Aplikasi Pengembangan Produk

Berdasarkan Gambar 4.14 dapat diketahui bahwa *tab main menu* memiliki alur yang paling banyak. Hal ini dikarenakan fungsi *tab main menu* sebagai penyaji berbagai menu yang ada pada aplikasi pengembangan produk.

4.5.2.4 Desain Form Input

Desain *Form input* bertujuan untuk merancang *user interface* pada proses *input* data dalam bentuk *form*. Proses *input* data dibagi menjadi dua yaitu proses *input* data produk dan proses *input* data fungsi dan alternatif. Berikut merupakan desain dari tiap *form input*.

The screenshot shows a software window titled 'Aplikasi Pengembangan Produk v.1'. It has a menu bar with 'Menu', 'Input Data', and 'Log out'. Below the menu is a 'Data Produk' table with columns 'No.' and 'ID Produk', containing one row with '1' and '1'. A modal form titled 'Data Produk' is open, with fields for 'ID Produk' (2), 'Nama Produk', and 'Tanggal Pembuatan' (28 December 2015). At the bottom of the modal are 'Batal' and 'Simpan' buttons.

Gambar 4.15 Desain *Form Input* Data Project Pengembangan Produk

The screenshot shows a software window titled 'Proses Penentuan Alternatif'. It has a menu bar with 'Data Fungsi', 'Diagram FAST', 'Solusi Alternatif', and 'Kombinasi'. Below the menu is a table with columns 'No.' and 'ID Fungsi'. A modal form titled 'Input Data Fungsi' is open, with fields for 'ID Produk' (1), 'Nama Produk' (Konveyor), 'Tanggal Pembuatan' (29-December-2015), 'ID Fungsi', 'Nama Fungsi', and three 'Alternatif' fields (1, 2, 3). At the bottom of the modal are 'Simpan', 'Batal', and 'Proses' buttons.

Gambar 4.16 Desain *Form Input* Fungsi

Pada Gambar 4.15 dan 4.16 menjelaskan desain awal *form input*. Desain ini berdasarkan kebutuhan data pada desain *database* di sub-bab sebelumnya, seperti data fungsi, alternatif 1 hingga alternatif 3 yang dimiliki oleh setiap fungsinya.

4.5.2.5 Desain Visual Output

Desain *Visual output* bertujuan untuk merancang bentuk *output* visual sesuai dengan kebutuhan *user*. Pada bab sebelumnya diketahui *output* dari program adalah *Morphology Chart* dan alternatif kombinasi desain Berdasarkan kebutuhan tersebut maka akan dirancang *output* yang sesuai dengan kebutuhan tersebut. Desain *output* visual akan dijelaskan pada Gambar 4.17.

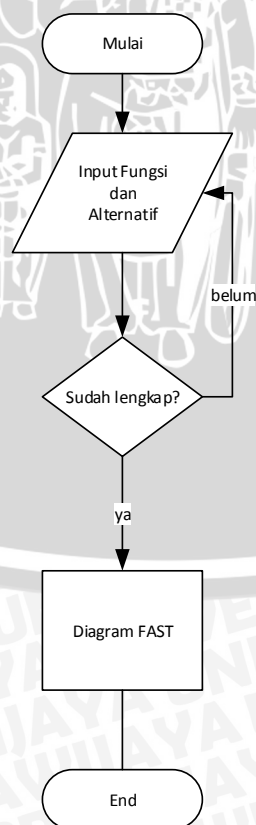
4.5.3 Desain Algoritma

Algoritma merupakan serangkaian urutan – urutan langkah logis untuk menyelesaikan suatu masalah yang disusun secara sistematis. Pada tahap desain algoritma ini, membahas tentang tahapan proses untuk menghasilkan output yang diperlukan dari sistem informasi yang dinyatakan dalam *pseudocode*. Berikut merupakan *pseudocode* untuk beberapa proses yang terjadi pada aplikasi pengembangan produk.

4.5.3.1 Desain Algoritma Diagram FAST

Algoritma pengerjaan Diagram FAST memiliki proses input data fungsi dan alternatif pada awal proses, setelah itu ada proses pengecekan ulang terhadap data tersebut, setelah data diinputkan maka akan keluar diagram FAST sesuai dengan data yang diinputkan sebelumnya, untuk lebih lengkapnya dapat dilihat pada penjabaran sebagai berikut:

1. Mulai
2. Siapkan data berupa technical response beserta alternatif
3. Cek kelengkapan data
4. Jika data belum lengkap maka masukan kembali data.
5. Jika data sudah lengkap maka terciptalah diagram FAST
6. Selesai

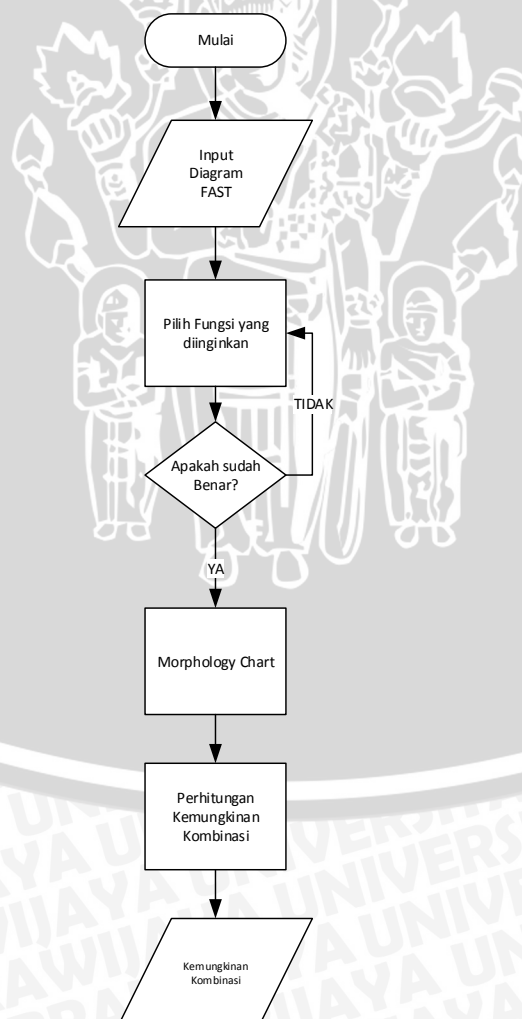


Gambar 4.20. Flowchart Algoritma dari Diagram FAST

4.5.3.2 Desain Algoritma *Morphology Chart*

Pada proses Morphology Chart, Input data diambil oleh data fungsi yang dipilih oleh user pada saat *windows* diagram FAST, setelah memilih fungsi mana saja yang akan digunakan maka terciptalah *morphology chart*, dan dari data *morphology chart* akan dihitung jumlah kombinasi alternatif didalam produk tersebut, algoritma Morphology Chart pada *prototype* aplikasi ini ialah sebagai berikut:

1. Mulai
2. Input Diagram FAST
3. Pilih fungsi sesuai dengan kebutuhan produk
4. Pengecekan Fungsi yang dipilih
5. Jika tidak maka cek maka kembali pilih fungsi yang sesuai
6. Jika iya maka terciptalah *morphology chart*
7. Hitung kemungkinan alternatif yang ada pada *morphology chart*
8. Selesai



Gambar 4.21. Flowchart Algoritma dari *Morphology Chart*

4.6 Implementasi

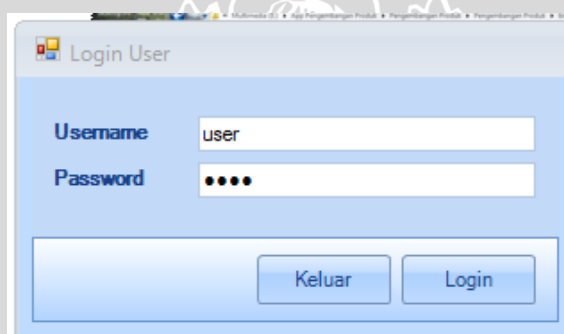
Implementasi adalah tahap pengaplikasian desain yang telah dirancang sebelumnya ke dalam *software* yang digunakan pada penelitian ini, *Microsoft Access*, Implementasi program bertujuan untuk memasukkan komponen dan mengatur kesesuaian antara program dan rancangan yang telah dibuat.

4.6.1 Implementasi User Interface

Tahap ini ditujukan agar pengguna dapat mengakses aplikasi pengembangan produk lebih mudah. Implementasi *user interface* dilakukan di tiap *windows* yang terlihat, dimana didasarkan pada analisa dan desain yang telah dibuat di tahap sebelumnya.

1. Windows Login

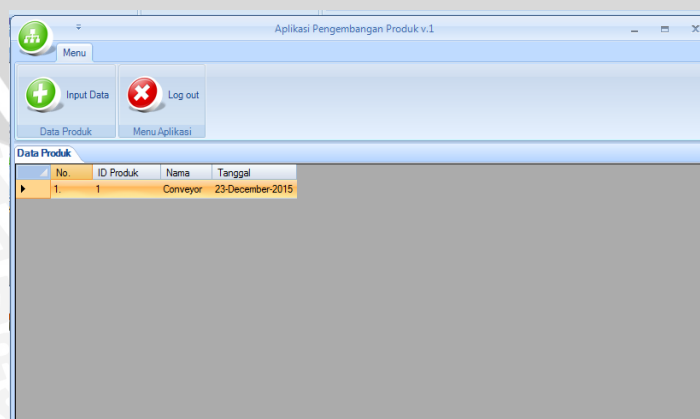
Windows login merupakan *windows* yang muncul sebelum menu utama dimana *user* harus memasukkan *username* dan *password* untuk dapat masuk ke dalam sistem.



Gambar 4.22 Menunjukkan Windows Login

2. Main Menu

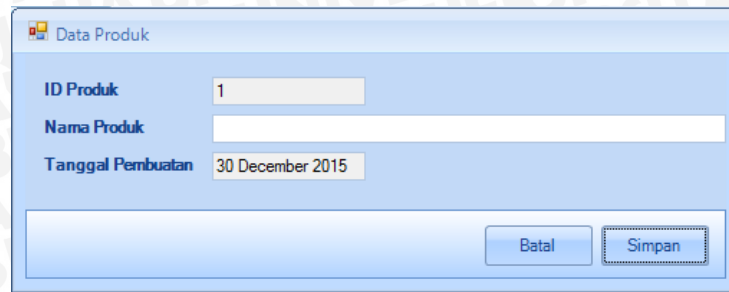
Main Menu merupakan *window* yang menampilkan berbagai pilihan fungsi program mulai dari perhitungan *input* data dan log out Gambar 4.22 menunjukkan *Windows Main Menu*



Gambar 4.23 Window Main Menu

3. Windows Input Data Project

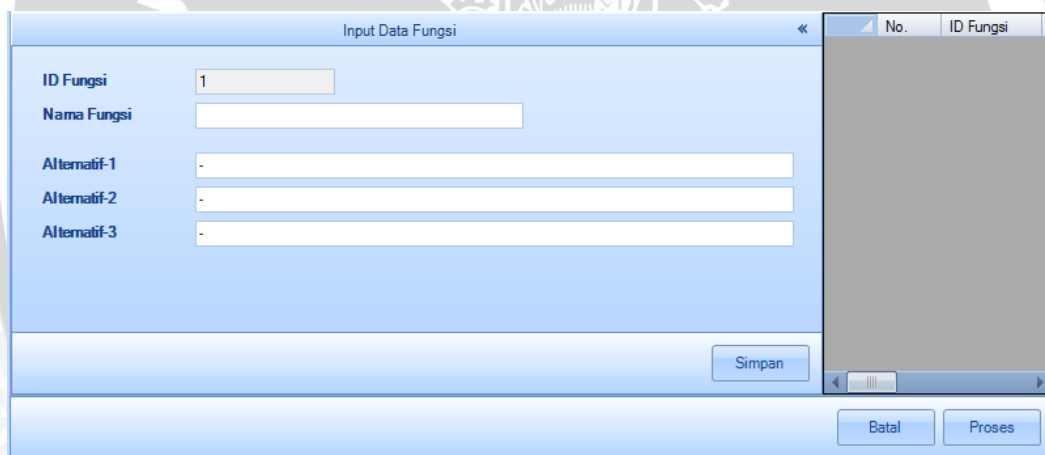
Window Input Data proyek merupakan window yang menampilkan fungsi input data keterangan proyek. Gambar 4.24 menunjukkan window Input Data Project.



Gambar 4.24 Window Input Data Project

4. Tab Input Data Fungsi dan Alternatif

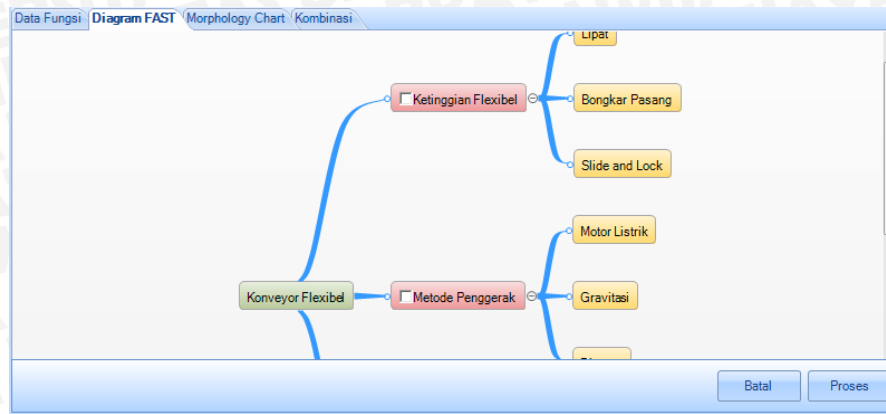
Tab Input Data fungsi dan Alternatif merupakan tab yang menampilkan fungsi input data fungsi dan harga alternatif. Pada tab ini akan ditampilkan form pengisian input data fungsi dan harga alternatif



Gambar 4.25 Tab Input Data Fungsi dan Harga Alternatif

5. Tab Diagram FAST

Tab Diagram FAST merupakan tab yang menampilkan diagram FAST. Pada tab ini akan ditampilkan Diagram FAST. Pada tab ini user dapat memilih fungsi yang akan digunakan digunakan berserta alternatif nya.



Gambar 4.26 Tab Diagram FAST

6. Tab Morphology Chart

Tab Morphology Chart merupakan tab yang menampilkan tabel data fungsi dan alternatif yang telah dipilih oleh user pada tab diagram fast sebelumnya.

Proses Penentuan Alternatif

ID Produk: 1
 Nama Produk: Conveyor
 Tanggal Pembuatan: 12-January-2016

Tab: Data Fungsi | Diagram FAST | **Morphology Chart** | Kombinasi

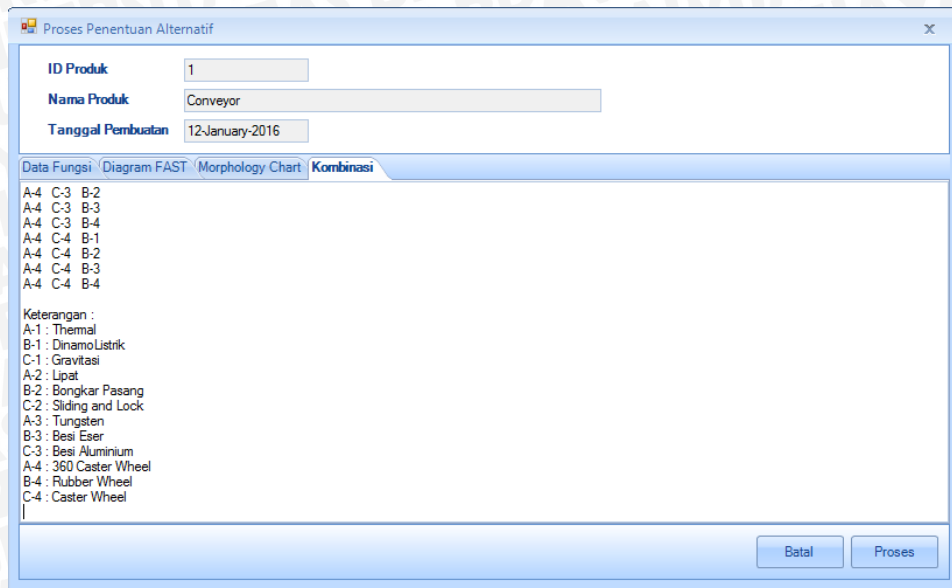
No.	Nama Fungsi	Alternatif-1	Alternatif-2	Alternatif-3
1.	Sumber tenaga ramah lingkungan	A-1 : Thermal	B-1 : Dinamo Listrik	C-1 : Gravitasi
2.	Flexibilitas Desain	A-2 : Lipat	B-2 : Bongkar Pasang	C-2 : Sliding and Lock
3.	Kuat dan ringan	A-3 : Tungsten	B-3 : Besi Eser	C-3 : Besi Aluminium
4.	Flexibilitas Pemindahan Conveyor	A-4 : 360 Caster Wheel	B-4 : Rubber Wheel	C-4 : Caster Wheel

Batal Proses

Gambar 4.27 Tab Morphology Chart

7. Tab Kombinasi Alternatif

Tab Kombinasi Alternatif merupakan tab yang menampilkan kombinasi dari alternatif yang telah ditampilkan pada tab morphology chart.



Gambar 4.28 Tab Kombinasi Alternatif

4.6.2 Implementasi Modul Program

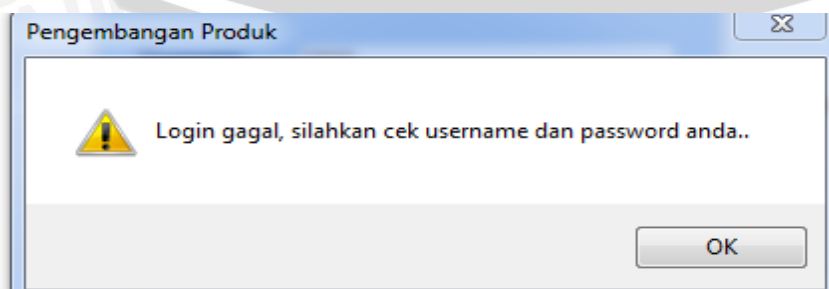
Berikut merupakan contoh beberapa *source code* untuk algoritma proses yang terjadi dalam program yang dibangun menggunakan *Microsoft Visual Studio*.

1. *Source code* untuk login

Cuplikan *syntax login user* ditampilkan pada Gambar 4.29 berikut. *Source code* selengkapnya ada di bagian Lampiran.

```
While reader.Read()
    If reader.HasRows Then
        ada = True
    End If
End While
reader.Close()
cnn.Close()
If (ada) Then
    FormMain.Show()
    Close()
    Exit Sub
Else
    MsgBox("Login gagal, silahkan cek username dan password anda..", MsgBoxStyle.Exclamation)

End If
TextBoxX2.Text = ""
TextBoxX1.Text = ""
TextBoxX1.Focus()
End Sub
```

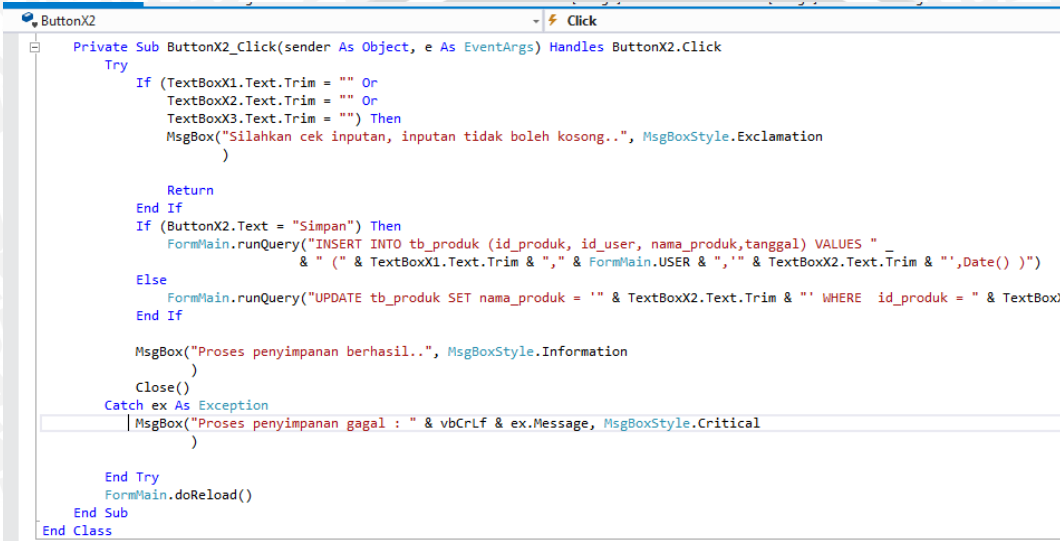


Gambar 4.29 Implementasi *syntax input login*

Pada Gambar 4.29 dapat diketahui bahwa *syntax* berhasil di implementasikan dan dapat di gunakan di program dengan baik. Sehingga user yang menggunakan produk ini hanyalah user yang telah terdaftar dalam database program.

2. *Source code* untuk *input* data fungsi dan alternatif

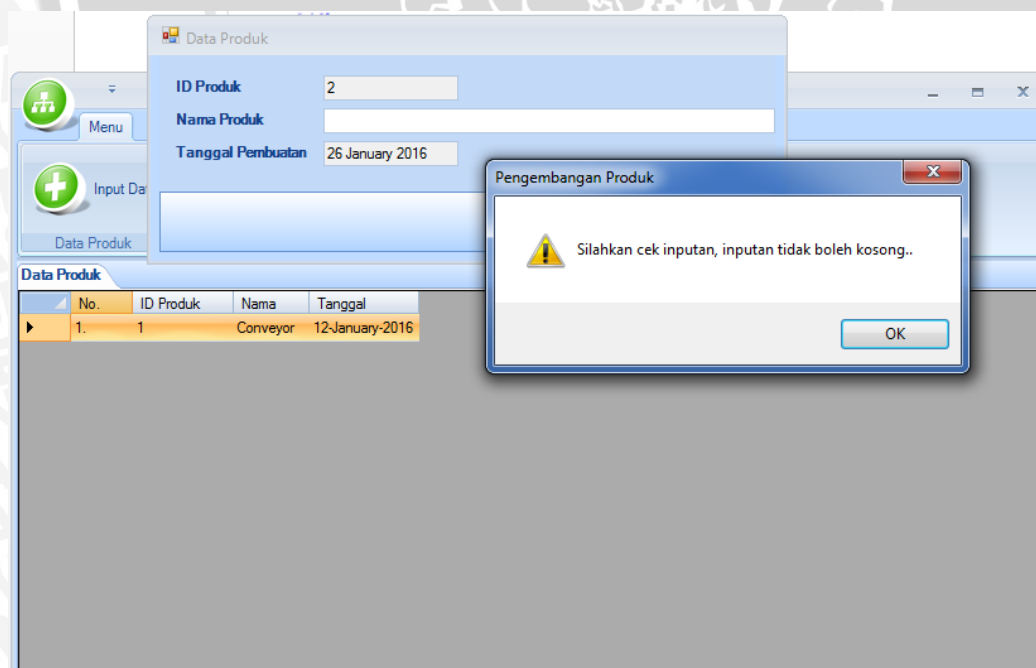
Gambar 4.30 berikut menunjukkan cuplikan *syntax* yang digunakan. *Source code* selengkapnya ada di bagian Lampiran.



```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Try
        If (TextBox1.Text.Trim = "" Or
            TextBox2.Text.Trim = "" Or
            TextBox3.Text.Trim = "") Then
            MsgBox("Silahkan cek inputan, inputan tidak boleh kosong..", MsgBoxStyle.Exclamation)
        End If
        Return
    End If
    If (Button2.Text = "Simpan") Then
        FormMain.runQuery("INSERT INTO tb_produk (id_produk, id_user, nama_produk,tanggal) VALUES " &
            & " (" & TextBox1.Text.Trim & "," & FormMain.USER & "," & TextBox2.Text.Trim & "','" & TextBox3.Text.Trim & "','" & Date() & "')"
    Else
        FormMain.runQuery("UPDATE tb_produk SET nama_produk = '" & TextBox2.Text.Trim & "' WHERE id_produk = '" & TextBox1.Text.Trim & "'")
    End If

    MsgBox("Proses penyimpanan berhasil..", MsgBoxStyle.Information)
    Close()
Catch ex As Exception
    MsgBox("Proses penyimpanan gagal : " & vbCrLf & ex.Message, MsgBoxStyle.Critical)
End Try
FormMain.doReload()
End Sub
End Class
```

Gambar 4.30 *Syntax Input Data Fungsi Dan Alternatif*



No	ID Fungsi	Nama Fungsi	Alternatif-1	Alternatif-2	Alternatif-3
1.	1	Sumber tenaga ramah lingkungan	Gravitasi	Dinamo Listrik	Thermal
2.	2	Flexibilitas Desain	Sliding and Lock	Bongkar Pasang	Lipat
3.	3	Kuat dan ringan	Besi Aluminium	Besi Eser	Tungsten
4.	4	Flexibilitas Pemindahan Conveyor	Caster Wheel	Rubber Wheel	360 Caster Wheel

Gambar 4.31 Implementasi *syntax* data fungsi dan alternatif

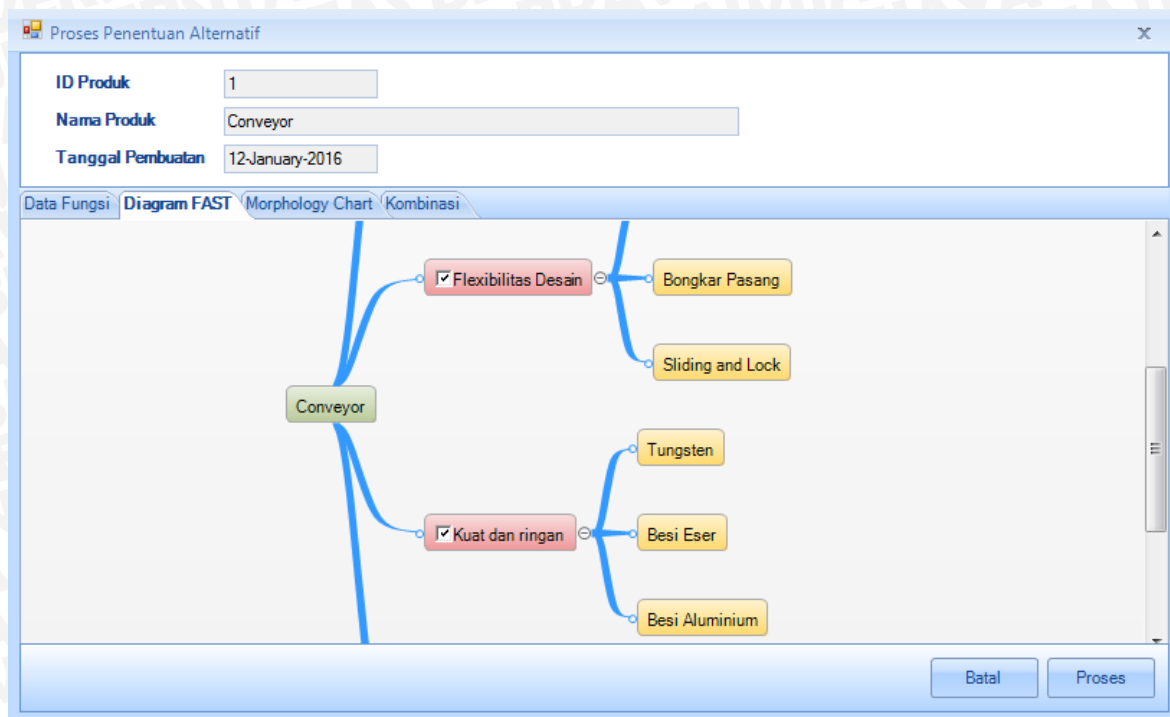
Pada Gambar 4.31 dapat kita lihat bahwa program dapat berfungsi dengan baik, dan data fungsi yang di masukan oleh *user* sesuai dengan pengelompokan *techincal response* pada fase pengumpulan data, dengan menyertakan alternatif yang telah diisi sesuai dengan hasil *brainstroming* dengan pihak terkait.

3. *Source code* untuk transformasi Diagram FAST

Gambar 4.32 berikut menunjukkan *syntax* yang digunakan untuk trasformasi Diagram FAST. *Source code* selengkapnya ada di bagian Lampiran.

```
Private Sub generateGrap()
    Me.treeGX1 = New DevComponents.Tree.TreeGX
    Me.projectNode = New DevComponents.Tree.Node
    Me.nodeConnector2 = New DevComponents.Tree.NodeConnector
    Me.nodeConnector1 = New DevComponents.Tree.NodeConnector
    Me.elementStyle1 = New DevComponents.Tree.ElementStyle

    CType(Me.treeGX1, System.ComponentModel.ISupportInitialize).BeginInit()
    Me.SuspendLayout()
    'treeGX1
    Me.treeGX1.AllowDrop = True
    Me.treeGX1.AutoScrollMinSize = New System.Drawing.Size(44, 22)
    'treeGX1.BackgroundColor
    Me.treeGX1.BackgroundColor2SchemePart = DevComponents.Tree.eColorSchemePart.BarBackground2
    Me.treeGX1.BackgroundColorSchemePart = DevComponents.Tree.eColorSchemePart.BarBackground
    Me.treeGX1.CommandBackColorGradientAngle = 90
    Me.treeGX1.CommandMouseOverBackColor2SchemePart = DevComponents.Tree.eColorSchemePart.ItemHotBackground2
    Me.treeGX1.CommandMouseOverBackColorGradientAngle = 90
    Me.treeGX1.Dock = System.Windows.Forms.DockStyle.Fill
    Me.treeGX1.ExpandLineColorSchemePart = DevComponents.Tree.eColorSchemePart.BarDockedBorder
    Me.treeGX1.Location = New System.Drawing.Point(0, 0)
    Me.treeGX1.Name = "treeGX1"
    Me.treeGX1.Nodes.AddRange(New DevComponents.Tree.Node() {Me.projectNode})
    Me.treeGX1.NodesConnector = Me.nodeConnector2
    Me.treeGX1.NodeStyle = Me.elementStyle1
End Sub
```



Gambar 4.32 Syntax dan Implementasi Diagram FAST

Berdasarkan Gambar 4.32, ditunjukkan bahwa syntax dari diagram FAST dapat bekerja dengan baik. Pada diagram FAST *user* dapat memilih fungsi sesuai dengan kebutuhan, dimana setiap fungsi membawa 3 alternatif. Seperti fungsi kuat dan ringan memiliki alternatif yaitu, tungsten, besi eser, dan besi Aluminium.

4. *Source code* untuk perhitungan optimal dengan *Morphology Chart* dan Pembangkitan Kombinasi

Gambar 4.33 berikut menunjukkan *source code* untuk proses *morphology chart source code* selengkapnya ada di bagian Lampiran.

```

FormProses
InitializeComponent

FormProses
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.ClientSize = New System.Drawing.Size(799, 458)
Me.Controls.Add(Me.TabControl1)
Me.Controls.Add(Me.PanelEx1)
Me.Controls.Add(Me.PanelEx2)
Me.DoubleBuffered = True
Me.EnableGlass = False
Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.Fixed3D
Me.MaximizeBox = False
Me.MinimizeBox = False
Me.Name = "FormProses"
Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
Me.Text = "Proses Penentuan Alternatif"
CType(Me.TabControl1, System.ComponentModel.ISupportInitialize).EndInit()
Me.TabControl1.ResumeLayout(False)
Me.TabControlPanel14.ResumeLayout(False)
Me.TabControlPanel13.ResumeLayout(False)
CType(Me.DTGrid_2, System.ComponentModel.ISupportInitialize).EndInit()
Me.ContextMenuStrip1.ResumeLayout(False)
Me.TabControlPanel12.ResumeLayout(False)
Me.TabControlPanel11.ResumeLayout(False)
Me.PanelEx3.ResumeLayout(False)
CType(Me.DTGrid, System.ComponentModel.ISupportInitialize).EndInit()
Me.ExpandablePanel1.ResumeLayout(False)
Me.PanelEx5.ResumeLayout(False)
Me.PanelEx1.ResumeLayout(False)
Me.PanelEx2.ResumeLayout(False)
Me.ResumeLayout(False)

```

Proses Penentuan Alternatif

ID Produk:

Nama Produk:

Tanggal Pembuatan:

Data Fungsi | Diagram FAST | **Morphology Chart** | Kombinasi

No.	Nama Fungsi	Alternatif-1	Alternatif-2	Alternatif-3
1.	Sumber tenaga ramah lingkungan	A-1 : Thermal	B-1 : Dinamo Listrik	C-1 : Gravitasi
2.	Flexibilitas Desain	A-2 : Lipat	B-2 : Bongkar Pasang	C-2 : Sliding and Lock
3.	Kuat dan ringan	A-3 : Tungsten	B-3 : Besi Eser	C-3 : Besi Aluminium
4.	Flexibilitas Pemindahan Conveyor	A-4 : 360 Caster Wheel	B-4 : Rubber Wheel	C-4 : Caster Wheel

Batal Proses

Gambar 4.33 Implementasi *Syntax Morphology Chart* dan pembangkitan kombinasi

Pada Gambar 4.33 ditunjukkan bahwa pada Proses Morphology Chart berhasil diimplementasikan. Berdasarkan data yang telah di olah user, pada perancangan *flexible roller conveyor* ini memiliki 4 Fungsi dan 4 alternatif karena pada proses diagram FAST keseluruhan fungsi dipilih oleh user. Sementara pada Proses pembangkitan kombinasi dapat diimplementasikan sesuai dengan *syntax* yang telah di rancang sebelumnya. Untuk hasil kombinasi lebih lengkap tersedia di Lampiran 3.

4.6.3 Validasi

Validasi menguji apakah fungsi *prototype* yang telah dirancang sudah sesuai dengan fungsi yang diharapkan dan telah merepresentasikan tujuan awal berdasarkan *user requirement* yang dijabarkan dalam *System Requirement Checklist* (SRC). Uji validasi dilakukan dengan menguji coba aplikasi kepada *user* bagian produksi PT. Flamboyan Jaya. Perbandingan hasil *prototype* sistem dengan *System Requirement Checklist* disajikan pada Tabel 4.15.

Berdasarkan hasil uji validasi Tabel 4.15, *prototype* aplikasi pengembangan produk sudah dapat memenuhi kebutuhan sistem yang dibutuhkan oleh *User*. *Prototype* dapat memberikan layanan *input* pada *user* untuk memasukkan data fungsi dan alternatif. Sistem dapat membangkitkan kemungkinan alternatif yang dimiliki oleh kombinasi fungsi, hal ini dapat dilihat pada Gambar 4.27 dengan ada nya 81 kemungkinan kombinasi dari fungsi yang dipilih, dengan menggunakan rumus kombinasi ialah

$$N = X^Y$$

N= kemungkinan kombinasi

X= banyaknya Fungsi

Y= banyak Alternatif

Maka $N = 3^4$, $N = 84$.

Sistem memiliki kemampuan dalam melakukan update data jika terdapat data baru. Sistem juga sudah dilengkapi sistem keamanan untuk masing-masing *user* dengan adanya *username* dan *password* untuk membatasi hak akses pengguna.

Tabel 4.13 Uji Validasi Aplikasi Pengembangan Produk Komponen *Input* dan *Output*

Komponen	Penjabaran	Kebutuhan pengguna yang dipenuhi
Input	User dapat memasukkan data berikut ini: - Produk: Produk yang sedang dikerjakan .. - Fungsi: <i>Technical response</i> dari HOQ yang telah dihimpun pada fase pertama perancangan produk - Alternatif : Alternatif yang dimiliki oleh fungsi dari <i>technical response</i>	User dapat memasukkan data produk, fungsi, dan alternatif. Dibuktikan dengan <i>screenshot</i> data produk, fungsi, dan alternatif.
Output	Sistem dapat memberikan <i>output</i> yang dibutuhkan, antara lain: 1. Diagram FAST . 2. <i>Morphology Chart</i> .	Sistem dapat memberikan <i>output</i> Diagram FAST dan <i>Morphology chart</i> . Dibuktikan dengan <i>screenshot</i> hasil <i>output</i> aplikasi.

Tabel 4.14 Uji Validasi Aplikasi Pengembangan Produk Komponen *Process*, *Performance* dan *Control*

Komponen	Penjabaran	Kebutuhan pengguna yang dipenuhi
Process	Sistem dapat membangkitkan kemungkinan jumlah kombinasi alternatif dari fungsi yang telah dimiliki oleh produk.	Sistem dapat membangkitkan kemungkinan jumlah kombinasi alternatif dari fungsi yang telah dimiliki oleh produk.. Dibuktikan dengan uji validitas di sub-bab 4.7.
Performance	Sistem memungkinkan dalam melakukan update data jika terdapat data baru.	Sistem memungkinkan dalam melakukan update data jika terdapat data baru karena sifat algoritma dalam aplikasi yang dinamis.
Control	User diberi <i>username</i> dan <i>password</i> untuk <i>login</i> sistem yang hanya dapat diakses oleh User itu sendiri.	Sistem menggunakan <i>username</i> dan <i>password</i> untuk <i>login</i> . Dibuktikan dengan <i>screenshot</i> login aplikasi.

4.6.4 Uji Prototype

Uji *prototype* bertujuan untuk mengetahui apakah *prototype* aplikasi sudah mengatasi kelemahan dan masalah dari segi *Performance*, *Information*, *Economy*, *Control*, *Efficiency*,

Services (PIECES). Prototype diujikan dengan sistem lama. Perbandingan sistem lama sebelum adanya *prototype* dan sistem baru dipaparkan pada Tabel 4.15 dan Tabel 4.16

Berdasarkan Tabel tersebut dapat diketahui bahwa sistem baru sudah dapat memperbaiki semua kelemahan sistem lama dari segi *performance*, *information*, *economy*, *control*, *efficiency* dan *service* (PIECES).

Tabel 4.15 Perbandingan Performa antara Sistem Lama dan Sistem Baru dengan analisa PIECES (*Performance* dan *Information*)

Jenis analisis	Sistem Lama	Sistem Baru
<i>Performance</i>	Sulitnya menelusuri informasi yang dibutuhkan oleh <i>User</i> alternatif dari technical response project yang pernah ada karena tidak ada dokumentasi mengenai hal tersebut	Sudah menggunakan <i>database</i> dalam penyimpanan berbasis komputer sehingga mudah untuk mencari data dan melihat data
<i>Information</i>	Informasi mengenai alternatif dan technical response tidak pernah ada dokumentasi, hanya tim pengembang yang memiliki pengetahuan lebih.	Data disimpan dalam <i>database</i> sehingga kecil kemungkinan data akan hilang atau tidak lengkap ketika diperlukan.

Tabel 4.16 Perbandingan Performa antara Sistem Lama dan Sistem Baru dengan analisa PIECES (*Economy*, *Control*, *Efficiency* dan *Services*)

Jenis analisis	Sistem Lama	Sistem Baru
<i>Economy</i>	Biaya yang dikeluarkan dan waktu yang dibutuhkan untuk menjalankan sistem tidak terlalu signifikan terhadap total biaya pengeluaran, namun berpotensi membesar seperti biaya administrasi untuk pembuatan dan penyimpanan dokumen secara manual dengan kertas dan bersifat sementara.	Biaya yang dikeluarkan dalam penggunaan program dijalankan di komputer minimal, karean hanya untuk biaya penggunaan listrik saja.
<i>Control</i>	Semua pihak dapat mengakses semua data karena data tidak dilengkapi sistem keamanan dan bersifat sementara.	Sistem telah dilengkapi <i>username</i> dan <i>password</i> untuk membatasi hak akses pengguna.
<i>Efficiency</i>	Sistem yang lama menentukan kombinasi alternatif dengan berdasarkan pengalaman. Memiliki kemungkinan kesalahan kombinasi alternatif yang mengakibatkan cacat dan juga waktu penentuan yang membutuhkan <i>brainstroming</i> .	Sistem yang baru menentukan kombinasi alternatif dari semua kemungkinan yang ada pada pengembangan produk, sehingga tidak membutuhkan waktu banyak untuk memikirkan kombinasi apa saja yang kemungkinan ada pada pengembangan produk
<i>Services</i>	Pelayanan akan kebutuhan informasi dari beberapa data berlangsung lama karena data belum terintegrasi menjadi satu sehingga sistem tidak fleksibel.	Sistem sudah terintegrasi dan fleksibel terhadap perubahan yang ada sehingga pelayanan akan kebutuhan informasi semua data lebih cepat dan akurat.

Sistem sudah terorganisir dengan baik untuk meminimalisasi redudansi data sehingga dapat menghemat biaya operasional serta waktu yang biasa dibutuhkan ketika mencari dokumen secara manual. Sistem juga mampu memberikan kombinasi alternatif secara lengkap kepada *user* sehingga dalam fase selanjutnya tidak membutuhkan waktu yang lama untuk kembali mengulang pada fase pengembangan konsep.



(Halaman ini sengaja dikosongkan)

