

LEMBAR PERSETUJUAN

**INTEGRASI SISTEM INFORMASI PENGENDALIAN KUALITAS
DENGAN *EXPERT SYSTEM* UNTUK MENELUSURI PENYEBAB
CACAT PRODUK**

(Studi Kasus PT. Yamaha Musical Products Indonesia)

**SKRIPSI
KONSENTRASI SISTEM INFORMASI MANAJEMEN INDUSTRI**

**Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik**



Disusun oleh :
ANGGUN PUSPITA ASRI
NIM. 0510670008-62

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Purnomo Budi Santoso, MSc., Ph.D.
NIP. 19530113 198303 1 003

Nasir Widha Setvanto, ST., MT.
NIP. 19700914 200501 1 00



**INTEGRASI SISTEM INFORMASI PENGENDALIAN KUALITAS
DENGAN *EXPERT SYSTEM* UNTUK MENELUSURI PENYEBAB
CACAT PRODUK
(Studi Kasus PT. Yamaha Musical Products Indonesia)**

**SKRIPSI
KONSENTRASI SISTEM INFORMASI MANAJEMEN INDUSTRI**

**Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik**



Disusun oleh :
ANGGUN PUSPITA ASRI
NIM. 0510670008-62

**KEMENTERIAN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2010**

LEMBAR PENGESAHAN

**INTEGRASI SISTEM INFORMASI PENGENDALIAN KUALITAS
DENGAN *EXPERT SYSTEM* UNTUK MENELUSURI PENYEBAB
CACAT PRODUK**

(Studi Kasus PT. Yamaha Musical Products Indonesia)

SKRIPSI

KONSENTRASI SISTEM INFORMASI MANAJEMEN INDUSTRI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh :
ANGGUN PUSPITA ASRI
NIM. 0510670008-62

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 11 Februari 2010

Skripsi 1

Skripsi 2

Ir. Mochamad.Choiri, MT.
NIP. 19540104 198602 1 001

Dr.,Ir. Pratikto, MMT.,
NIP. 19461110 198103 1 001

Komprehensif

Ir. Bambang Indrayadi, MT.
NIP. 19600905 198701 1 001

Mengetahui
Ketua Program Studi Teknik Industri

Nasir Widha Setyanto, ST.,MT.
NIP. 19700914 200501 1 001

PERNYATAAN ORISINALITAS JUDUL PENELITIAN SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam skripsi ini tidak terdapat karya ilmiah yang pernah diajukan orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam skripsi ini dapat dibuktikan terdapat unsur-unsur plagiat, saya bersedia skripsi ini dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No.20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, Februari 2010
Mahasiswa

Anggun Puspita Asri
NIM. 0510670008

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan hasil penelitian dengan judul INTEGRASI SISTEM INFORMASI PENGENDALIAN KUALITAS DENGAN *EXPERT SYSTEM* UNTUK MENELUSURI PENYEBAB CACAT PRODUK Studi Kasus PT. Yamaha Musical Products Indonesia dapat terselesaikan.

Pada kesempatan ini, penulis menyampaikan terima kasih kepada semua pihak yang telah membantu penulis dalam menyelesaikan laporan hasil penelitian ini, terutama kepada:

1. Bapak, ibu, kakak, adik dan seluruh keluarga, terima kasih atas kasih sayang serta bantuan doa dan dukungannya.
2. Ir. Purnomo Budi Santoso, MSc., Ph.D. selaku pembimbing utama dan Nasir Widha Setyanto, ST., MT. selaku pembimbing kedua dan Kaprodi Teknik Industri atas arahan dan bimbingannya hingga terselesaikannya penulisan laporan hasil penelitian ini.
3. Sahabat-sahabatku dan teman-teman Teknik Industri 2005 beserta adik-adik angkatan tercinta atas bantuan dan dukungannya.
4. Keluarga besar Laboratorium Komputer Teknik Mesin atas doa dan dukungannya.
5. Geng Pitoe: Dyah, Delvira, Vini, Ganis, Felysia, dan Meme, atas dukungan yang sangat berarti.
6. Teman-teman kos Watu Gong 1, kebersamaan kita tak akan terlupakan.
7. Serta semua pihak yang tak dapat disebutkan namanya satu persatu, yang telah membantu dalam penyelesaian laporan penelitian ini.

Penulis menyadari bahwa laporan hasil penelitian ini masih kurang dari sempurna. Akhir kata penulis berharap agar laporan hasil penelitian ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Malang, Februari 2010

Penulis

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	viii
RINGKASAN	ix
SUMMARY	x
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	4
1.4 Batasan Masalah	4
1.5 Tujuan Penelitian	4
1.6 Asumsi	4
1.7 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA	
2.1 Pengendalian Kualitas	6
2.1.1 Definisi Pengendalian Kualitas	6
2.1.2 <i>Statistical Quality Control</i>	6
2.1.3 Diagram Pareto	7
2.1.4 Peta Kontrol	7
2.1.5 Diagram Sebab Akibat	9
2.2 Sistem Informasi	9
2.3 <i>Expert System</i>	10
2.3.1 Elemen <i>Expert System</i>	11
2.3.2 Cara Kerja <i>Expert System</i>	12
2.3.3 Teknik Inferensi	12
2.4 Sistem Basis Data	13
2.4.1 Diagram <i>Entity-Relationship</i>	14
2.4.2 <i>Data Flow Diagram</i>	16

2.5 Microsoft Access.....	17
2.5.1 Pengenalan <i>Microsoft Access</i>	17
2.5.2 Penyimpanan Data oleh <i>Microsoft Access</i>	17
2.6 Integrasi <i>Expert System</i> dan Sistem Basis Data.....	18
2.7 Proses Produksi PT. Yamaha Musical Products Indonesia.....	19
2.7.1 Urutan Proses Produksi Pianika.....	19
2.7.2 Diagram Alir Proses Produksi Pianika.....	21

BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian.....	25
3.2 Penelitian Pendahuluan.....	25
3.3 Identifikasi Masalah.....	25
3.4 Tujuan Perancangan.....	25
3.5 Studi Literatur.....	26
3.6 Pengumpulan Data.....	26
3.7 Perancangan Sistem.....	26
3.8 Penarikan Kesimpulan dan Saran.....	27

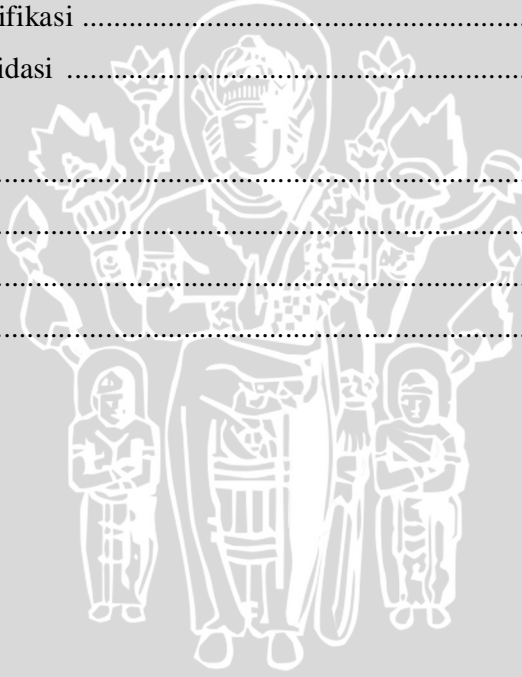
BAB IV PENGUMPULAN DATA

4.1 Pengantar.....	29
4.2 Pengetahuan Pakar.....	29
4.2.1 Proses Penyetelan Celah <i>Reed</i>	29
4.2.2 Proses Penyetelan Pengangkatan <i>Reed</i>	32
4.2.3 Proses Penyetelan <i>Reed</i>	33
4.3 Data Audit Produk.....	35

BAB V PERANCANGAN DAN PENGUJIAN SOFTWARE APLIKASI

5.1 Pengantar.....	39
5.2 Perencanaan.....	39
5.2.1 Spesifikasi <i>Software</i>	39
5.2.2 <i>List Entity</i>	40
5.3 Analisa Sistem.....	43
5.3.1 Diagram Konteks.....	43
5.3.2 <i>Data Flow Diagram</i>	44
5.3.3 DFD Level 1 Proses <i>Expert System</i>	45
5.3.4 <i>Entity Relationship Diagram</i>	46
5.3.5 Perancangan <i>Blok Diagram</i>	47

5.3.6 Perancangan <i>Dependency Diagram</i>	47
5.4 Desain Sistem	48
5.4.1 Perancangan <i>Database Fisik</i>	48
5.4.2 Perancangan <i>Decision Table</i>	52
5.4.3 Desain <i>Inference Engine</i>	53
5.5 Implementasi	53
5.5.1 Input Data Audit	53
5.5.2 <i>Statistical Quality Control</i>	54
5.5.3 Laporan/ <i>Report</i>	57
5.5.4 <i>Expert System</i>	59
5.6 Pengujian <i>Software</i>	60
5.6.1 Personel Yang Melakukan Pengujian	61
5.6.2 Uji Verifikasi	61
5.6.3 Uji Validasi	63
BAB VI PENUTUP	
6.1 Kesimpulan	66
6.2 Saran	66
DAFTAR PUSTAKA	67
LAMPIRAN	68



DAFTAR TABEL

No.	Judul	Halaman
Tabel 1.1	Prosentase Cacat Produk Pianika P32	2
Tabel 2.1	Notasi yang digunakan dalam diagram E-R	14
Tabel 2.2	Simbol-Simbol Pada DFD	16
Tabel 4.1	Audit <i>Finished Good</i> P32 Juli 2008	36
Tabel 4.2	Audit <i>Finished Good</i> P32 Agustus 2008	37
Tabel 4.3	Audit <i>Finished Good</i> P32 September 2008	38
Tabel 5.1	<i>List Entity</i>	40
Tabel 5.2.	Tabel P25data	49
Tabel 5.3.	Tabel P32data	49
Tabel 5.4.	Tabel P37data	49
Tabel 5.5.	Tabel Pegawai	50
Tabel 5.6.	Tabel <i>Input Expert System</i>	50
Tabel 5.7.	Tabel Identifikasi Awal	50
Tabel 5.8.	Tabel Mesin	50
Tabel 5.9.	Tabel Metode	50
Tabel 5.10.	Tabel Operator	51
Tabel 5.11.	Tabel Material	51
Tabel 5.12.	Tabel Proses	51
Tabel 5.13.	Tabel Solusi	51
Tabel 5.14.	Tabel <i>Defect</i>	51
Tabel 5.15.	Tabel <i>Decision</i>	51
Tabel 5.16.	Tabel <i>Rule Decision</i>	52
Tabel 5.17.	<i>Pseudo Code Inference Engine</i>	53
Tabel 5.18	Perhitungan Manual Atribut <i>Pareto Diagram</i>	63
Tabel 5.19.	<i>Rule / Aturan yang Bekerja Pada Uji Validasi</i>	64

DAFTAR GAMBAR

No.	Judul	Halaman
	Gambar 2.1 Contoh Diagram Pareto	7
	Gambar 2.2 Contoh Peta Kontrol	8
	Gambar 2.3 Contoh Diagram Sebab-Akibat	9
	Gambar 2.4 Hubungan Data dan Informasi	10
	Gambar 2.5. Elemen <i>Expert System</i>	11
	Gambar 2.6 Diagram Blok Umum <i>Expert System</i>	12
	Gambar 2.7 Contoh relasi satu ke satu	15
	Gambar 2.8 Contoh relasi satu ke banyak	15
	Gambar 2.9 Contoh relasi banyak ke banyak	15
	Gambar 2.10 Gambar Tampilan <i>Microsoft Access</i>	18
	Gambar 2.11 <i>Plate</i>	20
	Gambar 2.12. <i>Reed</i>	20
	Gambar 2.13. Diagram Alir Pada Proses <i>Reedplate</i>	22
	Gambar 2.14. Diagram Alir Pada <i>Initial Process</i>	23
	Gambar 2.15. Diagram Alir Pada <i>Finish Process</i>	24
	Gambar 3.1. <i>Flowchart</i> Metodologi Penelitian	28
	Gambar 4.1 Sudut Pandang Cara Melihat <i>Reed</i> Pada <i>Reed Plate</i> yang Benar	30
	Gambar 4.2 Sudut Pandang Cara Melihat <i>Reed</i> Pada <i>Reed Plate</i> yang Salah	30
	Gambar 4.3 Cara memegang <i>Spanner</i> dengan Benar	31
	Gambar 4.4 Cara memegang <i>Spanner</i> yang Salah	31
	Gambar 4.5 Penempatan <i>Spanner</i> Pada Kepala <i>Reed</i> yang Benar	31
	Gambar 4.6 Cara Perbaikan <i>Reed</i> yang Benar	32
	Gambar 4.7 Cara Perbaikan <i>Reed</i> yang Salah	32
	Gambar 4.8 Posisi Hera yang Benar	33
	Gambar 4.9 Posisi Hera yang Salah	33
	Gambar 4.10 Posisi Pengikiran dengan Arah Kikir Lurus	34
	Gambar 4.11 Posisi Pengikiran dengan Arah Kikir Miring	34
	Gambar 4.12 Hasil Pengikiran yang Lurus / Rata	34
	Gambar 4.13 Hasil Pengikiran yang Miring	35
	Gambar 5.1 Diagram Konteks Aplikasi	43
	Gambar 5.2 DFD Level 0 SIPKES	44
	Gambar 5.3 DFD Level 1 Proses <i>Expert System</i>	45

Gambar 5.4 <i>Entity Relationship Diagram</i> SIPKES	46
Gambar 5.5 Blok Diagram <i>Expert System</i>	47
Gambar 5.6 <i>Dependency Diagram</i> Proses <i>Expert System</i>	48
Gambar 5.7 <i>Form</i> Audit Data Pianika P32	54
Gambar 5.8 <i>Form</i> SQC	55
Gambar 5.9 <i>Form</i> Pareto Diagram Pianika P32	55
Gambar 5.10 <i>Form</i> Peta Kontrol Pianika P32	56
Gambar 5.11 <i>Fishbone</i> Diagram untuk cacat produk Biri	56
Gambar 5.12 <i>Form</i> Laporan Pianika	57
Gambar 5.13 <i>Form</i> Laporan Pianika Bulanan	58
Gambar 5.14 <i>Form</i> Laporan Pianika Tahunan	58
Gambar 5.15 <i>Form</i> Input <i>Expert System</i>	59
Gambar 5.16 Hasil <i>Expert System</i>	60
Gambar 5.17 Hasil Uji Verifikasi Peta Kontrol	61
Gambar 5.18 Hasil Uji Verifikasi <i>Pareto Diagram</i>	62
Gambar 5.19 <i>Input</i> Data Uji Validasi <i>Expert System</i>	65
Gambar 5.20 Hasil Uji Validasi <i>Expert System</i>	65

DAFTAR LAMPIRAN

No.	Judul	Halaman
	Lampiran 1. Manual Prototipe	68



RINGKASAN

Anggun Puspita Asri, Program Studi Teknik Industri, Jurusan Teknik Mesin, Fakultas Teknik Universitas Brawijaya, Februari 2010, *Integrasi Sistem Informasi Pengendalian Kualitas Dengan Expert System Untuk Menelusuri Penyebab Cacat Produk*, Dosen Pembimbing : Ir. Purnomo Budi Santoso, Msc., Ph.D. dan Nasir Widha Setyanto, ST., MT.

Permasalahan yang dihadapi perusahaan PT. YMPI ini adalah masih terdapat cacat produk pada tiap produksinya yang menyebabkan kualitas produksi menjadi menurun, sistem pengendalian kualitas di PT. YMPI yang masih menggunakan sistem manual menyebabkan ketidakefisienan waktu dan terbatasnya jumlah tenaga pakar yang dapat mengidentifikasi masalah kecacatan produk, sehingga operator memerlukan waktu yang cukup lama (30 – 40 hari) untuk pemeriksaan cacat produk dan memperoleh solusi yang tepat.

Pada penelitian ini dilakukan proses perancangan prototipe *software* sistem pengendalian kualitas dan *expert system* atau yang disebut SIPKES. Aplikasi ini dikembangkan dalam lingkungan model relasional dengan menggunakan *Microsoft Access 2007*. *Data Flow Diagram* (DFD) digunakan untuk mendeskripsikan proses aliran data yang ada dalam aplikasi yang dirancang. Untuk menggambarkan pemodelan data atau desain basis data digunakan model *entity relationship diagram* yang selanjutnya ditransformasikan ke dalam bentuk basis data fisik. Basis pengetahuan didapat dari akuisisi pengetahuan dan direpresentasikan melalui *fishbone diagram*. Pembuatan *inference engine* dengan menggunakan bahasa pemrograman *Visual Basic for Application* dengan memanfaatkan teknologi *Active Data Object* (ADO) dan SQL.

Hasil dari penelitian ini adalah berupa prototipe aplikasi integrasi sistem pengendalian kualitas dengan *expert system* yang dapat digunakan pada divisi produksi. Dengan fasilitas yang diberikan untuk pemakai dan pakar, memungkinkan baik pemakai maupun pakar untuk menggunakan sistem ini sesuai kebutuhannya masing-masing. Pemakai diberi kemudahan untuk mendapatkan solusi atas cacat produk yang terjadi berdasarkan data-data yang telah dipilih pada proses penelusuran, yang terdiri dari identifikasi awal, kerusakan mesin, kesalahan metode, kesalahan operator, dan kondisi material. Pengendalian kualitas menggunakan *Statistical Quality Control* antara lain: *Pareto Diagram*, Peta Kontrol np, dan *Fishbone Diagram*. Sedangkan pakar diberi kemudahan dalam manajemen sistem, baik proses tambah, hapus maupun *update* data terkait.

Kata kunci: sistem informasi pengendalian kualitas, *expert system*

SUMMARY

Anggun Puspita Asri, Majoring in Industrial Engineering, Department of Mechanical Engineering, Faculty of Engineering Brawijaya University, February 2010, *Integration of Information System of Quality Control With Expert System To Trace cause of Product's Defects*, Academic Counselors : Ir. Purnomo Budi Santoso, Msc., Ph.D. dan Nasir Widha Setyanto, ST., MT.

The problem faced by PT.YMPI are the existence of product's defects in its production that can decrease quality of production. Quality control system of PT. YMPI which use manual system can cause inefficient in time and limited amount of expert people who can indentifying the problem that caused by product's defects, so operator need sufficient time (30 – 40 days) to inspection the product's defect and get the right solutions.

Process design of software prototype of quality control system and expert system or called SIPKES in this research. The application develop in relational model environment used Microsoft Access 2007. Data Flow Diagram (DFD) used to describe data flow process in the application. To depict of data's model or design of database used entity relationship diagram model later on transformed into physical database. Knowledge base got from knowledge acquisition and represented through fishbone diagram. Making of inference engine by using Visual Basic for Application by exploiting technology Active Data Object (ADO) dan SQL.

The result of this research is application prototype of integration of quality control information system with expert system that can be used in production division. With the facility that given for user and expert people, enabling both of them to use this system according to each requirement. User given the easy way to get the solutions for product's defects that happened based on the chosen data in trace process, consisted of early identifications, machine damage, method mistake, operator mistake, and material conditions. Quality control system use Statistical Quality Control, such as: Pareto Diagram, np Control Chart, and Fishbone Diagram. While expert given the amenity in system management, like to add, erase and also update the related data.

Key word: Information system of quality control, expert system

BAB I

PENDAHULUAN

1.1 Latar Belakang

Persaingan dalam dunia industri yang semakin ketat menuntut pelaku industri atau perusahaan untuk terus meningkatkan kualitas produknya. Hal tersebut bisa dicapai dengan pengendalian kualitas proses produksi yang baik. Tanpa adanya pengendalian kualitas proses produksi yang baik, maka produk yang dihasilkan tidak sesuai dengan spesifikasi yang diharapkan sehingga akan merugikan produsen maupun konsumen. Salah satu hal yang benar-benar harus diperhatikan untuk menjaga kepercayaan konsumen adalah masalah kualitas produk yang dihasilkan.

PT. Yamaha Musical Products Indonesia atau selanjutnya disebut PT.YMPI adalah jenis perusahaan manufaktur yang bergerak di bidang produksi alat musik. Salah satu produk yang dihasilkan adalah Pianika. Dengan kerjasama antara pekerja-pekerja terampil Indonesia, PT. YMPI menghasilkan produk dengan kesatuan yang baik dan tepat bagi industri alat musik yang menekankan pada kualitas yang tinggi, baik dari segi proses pembuatan maupun kualitas hasil produksi. Perusahaan ini memproduksi barang berdasarkan pesanan atau *order* dari konsumen, hal ini berarti konsumen menentukan jenis produk yang dipesan sekaligus spesifikasi-spesifikasi kualitas produk yang dipesannya. Untuk mempertahankan kepercayaan konsumen terhadap kinerja perusahaan, maka kualitas produk harus selalu dijaga dan ditingkatkan agar sesuai dengan kriteria yang telah ditetapkan.

Sistem pengendalian kualitas yang berjalan selama ini kurang baik, dimana beberapa cacat produksi yang sangat mempengaruhi kualitas Pianika masih ditemui di PT. YMPI. Cacat tersebut berakibat pada ketidaksesuaian antara produk yang dihasilkan dengan kriteria yang telah ditetapkan. Kecacatan yang terjadi pada proses produksi Pianika berbentuk cacat atribut, antara lain oktaf aneh, biri, reed tinggi, reed rendah, bunyi tersumbat, dan tidak bunyi. Berdasarkan data yang diperoleh dari PT. YMPI, jumlah total cacat produk rata-rata tiap bulannya adalah sekitar 0,9 % dari total produksinya. Cacat produksi tersebut harus dikurangi bahkan sedapat mungkin dihilangkan untuk meningkatkan kualitas produk.

Tabel 1.1 Prosentase Cacat Produk Pianika P32

Periode	Jumlah cacat produk	Total produksi	Prosentase
Juli 2008	415	50880	0.008
Agustus 2008	362	43680	0.008
September 2008	454	46200	0.010
		Rata-rata	0.009

Sumber: Laporan Perolehan Harian PT.YMPI, 2008

Perusahaan sendiri telah melakukan penelitian untuk mengetahui penyebab terjadinya cacat produk, sehingga kecacatan produk dapat diminimalkan. Pada kenyataannya, ketika cacat tersebut terjadi kembali, perusahaan tidak dapat menangani secara cepat, karena perlu mengkaji ulang penyebab serta mencari solusi yang tepat agar masalah tersebut dapat diatasi. Hal ini tentu saja memakan waktu yang cukup lama tanpa kepakaran tentang analisis kecacatan produk. Waktu yang biasa diperlukan untuk setiap pemeriksaan adalah sekitar 30 – 40 hari. Yang menjadi masalah adalah terbatasnya jumlah tenaga pakar yang dimiliki oleh perusahaan. Di samping itu, analisis kecacatan produk ini juga merupakan jenis pekerjaan yang memakan banyak waktu (*time consuming*). Oleh karena itu, diperlukan suatu solusi yang dapat mengatasi permasalahan ini.

Selain itu, sistem pengendalian kualitas yang digunakan masih menggunakan cara manual, yaitu dengan pencatatan pada kertas-kertas dan komputer hanya berfungsi sebagai mesin tik. Tidak ada sistem basis data yang dapat menampung data-data yang cukup banyak. Hal ini tentu saja sangat berpengaruh dalam pengelolaan data, pencarian data, dan penyimpanan data yang akan memakan banyak waktu dan tentu saja tidak efisien.

Perkembangan Teknologi Informasi, khususnya teknologi komputer, yang sedemikian pesatnya telah menghasilkan sebuah bidang yang disebut *Expert System* (Sistem Pakar) yang merupakan salah satu cabang dari *Artificial Intelligence* (Kecerdasan Buatan). *Expert System* mampu mengotomatisasikan pekerjaan-pekerjaan rutin yang memerlukan tenaga pakar sehingga awam pun mampu melaksanakan pekerjaan tersebut. Berdasarkan karakteristik dan kemampuan yang dimilikinya inilah, maka *Expert System* cocok digunakan untuk memecahkan permasalahan analisis kecacatan produk di atas, tetapi *Expert System* masih perlu didukung oleh teknologi lain

yang mampu menutupi kelemahannya, dan salah satu teknologi tersebut adalah Sistem Basis Data.

Integrasi *Expert System* dan sistem *database* akan sangat menarik karena kedua teknologi ini dapat saling melengkapi, keduanya dapat digunakan secara bersamaan untuk memecahkan bagian-bagian yang berbeda pada suatu masalah. *Expert System* memiliki kecerdasan setingkat pakar namun tidak mempunyai katalog atau sistem basis data, sedangkan sistem *database* memiliki kemampuan mengelola data dan informasi namun tidak mempunyai kecerdasan (*Expert System*).

Salah satunya adalah dengan diterapkannya pengintegrasian tersebut untuk memberikan solusi serta informasi penyebab kecacatan produk, sehingga permasalahan dapat diidentifikasi sedini mungkin. Sistem bekerja pada basis data untuk mengolah data kecacatan produk, sehingga diketahui cacat apa saja yang terjadi, kemudian melalui *Expert System* dapat diketahui cacat yang terjadi, proses kritis solusi yang tepat.

Berdasarkan uraian di atas, maka dipandang sangat perlu untuk merancang dan membuat *software* yang merupakan integrasi *Expert System* dan sistem *database* untuk membantu operator produksi dalam mengidentifikasi permasalahan kecacatan produk secara cepat dan tepat.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, maka masalah PT. YMPI dapat diidentifikasi sebagai berikut:

1. Masih terdapat cacat produk yang dapat menurunkan kualitas produk PT. YMPI
2. Terbatasnya jumlah tenaga pakar yang dapat mengidentifikasi masalah kecacatan produk.
3. Proses pemeriksaan atau pengkajian ulang penyebab kecacatan serta pencarian solusi merupakan pekerjaan yang memakan banyak waktu sehingga dapat mengakibatkan ketidak efisienan.
4. Sistem pengendalian kualitas yang diterapkan masih menggunakan cara manual.

1.3 Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah diatas, maka masalahnya dapat dirumuskan:

Bagaimana merancang sistem informasi pengendalian kualitas terintegrasi dengan *Expert System* untuk membantu menelusuri cacat produk, sehingga didapatkan penyebab serta solusi secara cepat.

1.4 Batasan Masalah

Adapun permasalahan dan pembahasan yang akan dianalisa PT.YMPI hanya terbatas pada masalah:

1. Perancangan *software* aplikasi hanya dalam bentuk sebuah prototipe
2. Implementasi dari prototipe berbasis komputer yang dirancang menggunakan *Microsoft Access*.
3. Desain integrasi *Expert System* dan Sistem Basis Data hanya sebatas untuk memberikan informasi penyebab kecacatan produk serta solusi yang direkomendasikan atas satu jenis produk tertentu yang merupakan salah satu produk yang diproduksi perusahaan, yaitu pianika.

1.5 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Mengintegrasikan Sistem Informasi Pengendalian Kualitas dan *Expert System* ke dalam sebuah prototipe aplikasi untuk membantu divisi produksi dalam perusahaan untuk mendapatkan solusi tentang cacat produk secara cepat dan tepat.
2. Menggantikan sistem pengendalian kualitas secara manual menjadi sistem pengendalian kualitas berbasis komputer.

1.6 Asumsi

Pada penelitian ini terdapat beberapa asumsi, antara lain:

1. Faktor lain yang tidak tercantum dalam penelitian ini dianggap tidak ada pengaruh yang signifikan.
2. Proses pengambilan data untuk penelitian dilakukan pada saat proses produksi berjalan dengan kondisi normal.

1.7 Manfaat Penelitian

Dari penulisan skripsi ini diharapkan mendapat manfaat sebagai berikut:

1. Bagi pihak perusahaan, dapat menyediakan kepakaran untuk mengidentifikasi permasalahan kecacatan produk yang terjadi secara cepat yang dapat digunakan dimana saja dan kapan saja sehingga dapat meningkatkan kualitas produk dan pengelolaan data kualitas lebih mudah dengan adanya sistem informasi pengendalian kualitas
2. Bagi penulis dan khalayak
 - a. Mengetahui bagaimana metode pengembangan *Expert System* dengan menggunakan *Microsoft Access*
 - b. Dapat mengaplikasikan pengintegrasian *Expert System* dan Sistem Basis Data walaupun hanya dalam bentuk prototipe untuk membantu mengatasi permasalahan kecacatan produk.
 - c. Memperoleh informasi tentang penyebab-penyebab yang menimbulkan cacat produk serta solusi yang direkomendasikan untuk mengatasi cacat produk tersebut.
3. Bagi pihak akademis dapat dijadikan sebagai referensi dalam menerapkan ilmu pengetahuan dan metodologi yang serupa untuk kemampuan penerapan ilmu pengetahuan dan teknologi yang telah diperoleh, serta dapat dijadikan pedoman dalam penelitian sejenisnya.

BAB II

TINJAUAN PUSTAKA

2.1. Pengendalian Kualitas

2.1.1 Definisi Pengendalian Kualitas

Pengendalian kualitas adalah aktivitas keteknikan dan manajemen, yang dengan aktivitas itu diukur ciri-ciri kualitas produk, membandingkan dengan spesifikasi atau persyaratan, dan mengambil tindakan penyehatan yang sesuai apabila ada perbedaan antara penampilan yang sebenarnya dan yang standar (Douglas C. Montgomery, 1991:3).

Dalam dokumen *ISO 8420 Quality Vocabulary* disebutkan bahwa pengertian mutu atau kualitas adalah keseluruhan corak dan karakteristik dari suatu produk atau jasa yang timbul atas kemampuannya untuk memuaskan kebutuhan pemakai atau konsumen.

2.1.2 *Statistical Quality Control*

Statistical Quality Control (SQC) yang berasal dari bahasa Inggris dan dikembangkan di Amerika dengan melalui suatu organisasi yang bernama *American for Quality Control* dibentuk tahun 1946. Organisasi ini mengembangkan penggunaan teknik pengendalian kualitas untuk segala macam produk atau jasa. Organisasi tersebut menyelenggarakan sejumlah komperensi, penerbitan teknik dan program latihan dalam jaminan kualitas.

Montgomery (1990:18) mengatakan SQC merupakan metode statistik dalam sisi pengambilan keputusan tentang suatu proses atau populasi berdasarkan pada suatu analisis informasi yang terkandung di dalam suatu sampel dari populasi itu.

Dalam hal ini *Statistical Quality Control* mampu digunakan untuk mendapatkan proses yang stabil dan memperbaiki kemampuan dan mengurangi variasi yang ada.

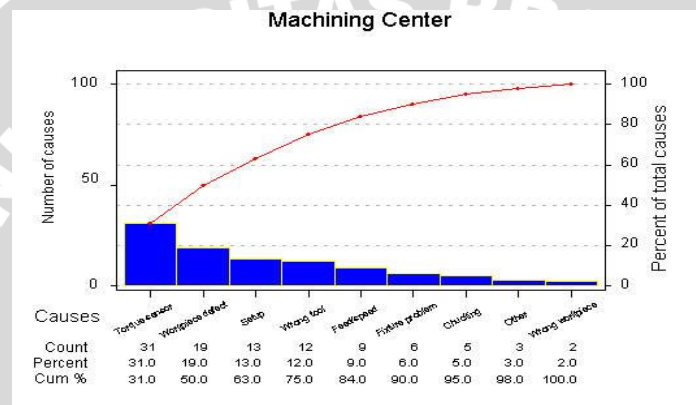
Ada tujuh alat utama dalam *Statistical Quality Control*:

1. Histogram
2. Lembar Periksa (*Check Sheet*)
3. Diagram *Pareto*
4. Diagram Sebab Akibat
5. Diagram *Scater*
6. *Control Chart* (Peta kendali)

7. Flowchart

2.1.3 Diagram Pareto

Menurut Gaspersz (2003), diagram pareto adalah grafik batang yang menunjukkan masalah berdasarkan urutan banyaknya kejadian. Masalah yang paling banyak terjadi ditunjukkan oleh grafik batang pertama yang tertinggi serta ditempatkan pada sisi paling kiri dan seterusnya sampai masalah yang paling sedikit terjadi ditunjukkan oleh grafik batang terakhir yang terendah serta ditempatkan pada sisi paling kanan.



Gambar 2.1 Contoh Diagram Pareto
Sumber: Kolarik, 1999

2.1.4 Peta Kontrol

Gaspersz (2003) berpendapat bahwa peta kontrol adalah peta yang digunakan untuk mencapai suatu keadaan terkendali, yaitu ketika semua nilai rata-rata dan kisaran sub kelompok berada dalam batas-batas pengendalian.

Pada dasarnya, peta-peta kontrol dipergunakan untuk :

1. Menentukan apakah suatu proses berada dalam pengendalian atau variasi penyebab-khusus menjadi tidak ada lagi dalam proses.
2. Memantau proses terus menerus sepanjang waktu agar proses tetap stabil secara statistik dan hanya mengandung variasi penyebab umum.
3. Menentukan kemampuan proses (*process capability*). Setelah proses berada dalam pengendalian, batas-batas variasi proses dapat ditentukan.

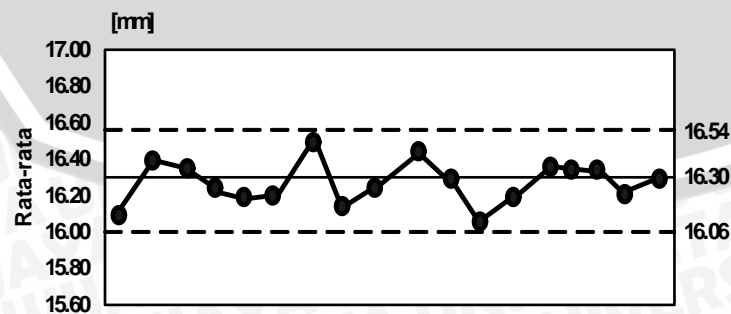
Peta kontrol dapat dibagi menjadi dua jenis, peta kontrol atribut dan peta kontrol variabel. Karakteristik kualitas yang dapat diukur dan dinyatakan secara kuantitatif

dinamakan variabel, sedangkan kualitas yang dinilai sebagai sesuai atau tidak sesuai (cacat) dinamakan atribut. Peta kontrol memberikan informasi tentang kemampuan proses, nilai parameter proses yang penting, dan stabilitas terhadap waktu sehingga memberikan taksiran kemampuan proses. Informasi ini sangat berguna bagi perancangan produk dan proses. Pengertian atribut dalam pengendalian kualitas berkaitan dengan karakteristik kualitas yang dapat digolongkan atas baik (diterima) dan cacat (ditolak). Beberapa macam peta kontrol atribut yaitu (Feigenbsum, 1994):

1. Peta kontrol p (p chart), yaitu peta kontrol untuk fraksi defektif (fraction rejected fraction nonconforming).
2. Peta kontrol np (np chart), yaitu peta kontrol untuk jumlah item yang tidak sesuai (number of nonconforming).
3. Peta kontrol c (c chart), yaitu peta kontrol untuk jumlah ketidaksesuaian (number of nonconformities)
4. Peta kontrol u (u chart), yaitu peta kontrol untuk jumlah ketidaksesuaian per unit (number of nonconformities per unit).

Setiap peta kontrol pada dasarnya memiliki kelengkapan sebagai berikut :

1. Garis Tengah (*Central Line*) yang biasa dinotasikan sebagai CL.
2. Sepasang batas kontrol (*Control Limits*) yaitu :
 - a. Batas kontrol yang ditempatkan di atas garis tengah yang dikenal sebagai batas kontrol atas (*Upper Control Limit*) dan dinotasikan sebagai UCL
 - b. Batas kontrol yang ditempatkan di bawah garis tengah yang dikenal sebagai batas kontrol bawah (*Lower Control Limit*) dan dinotasikan sebagai LCL.
3. Tebaran nilai-nilai karakteristik kualitas yang menggambarkan keadaan proses.



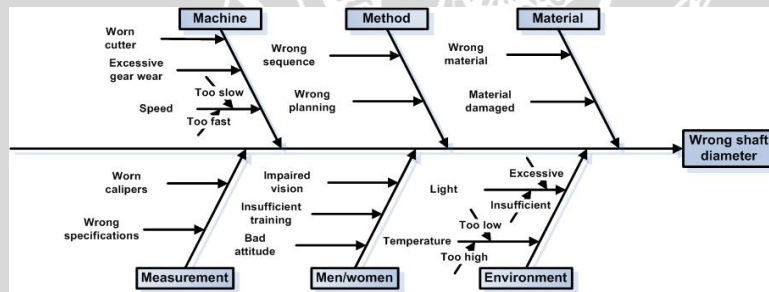
Gambar 2.2 Contoh Peta Kontrol
Sumber: Kolarik, 1999

2.1.5 Diagram Sebab-Akibat (Diagram Tulang Ikan)

Gaspersz (2003) menyatakan bahwa diagram sebab-akibat adalah suatu diagram yang menunjukkan hubungan antara sebab dan akibat. Berkaitan dengan pengendalian proses statistikal, diagram sebab akibat dipergunakan untuk menunjukkan faktor-faktor penyebab (sebab) dan karakteristik kualitas (akibat) yang disebabkan oleh faktor-faktor penyebab itu. Diagram sebab-akibat ini sering juga disebut sebagai diagram tulang ikan (*fishbone diagram*) karena bentuknya seperti kerangka ikan, atau diagram Ishikawa (*Ishikawa's Diagram*) karena pertama kali diperkenalkan oleh Prof. Kaoru Ishikawa dari Universitas Tokyo pada tahun 1943.

Langkah-langkah dalam pembuatan diagram sebab akibat secara keseluruhan adalah (Montgomery, 2001:122):

1. Menentukan karakteristik mutu yang akan dicari faktor-faktor penyebabnya.
2. Menggambarkan faktor-faktor utama penyebab ketidaksesuaian dengan menggambarkan garis panah menjadi garis utama
3. Dari faktor-faktor utama dicari sub faktor yang menyebabkan cacat. Sub faktor ini digabungkan pada faktor utama yang berkaitan dengan faktor tersebut.



Gambar 2.3 Contoh Diagram Sebab-Akibat
Sumber: Grant, 1999

2.2. Sistem Informasi

Definisi sistem informasi adalah kumpulan informasi didalam sebuah basis data menggunakan model dan media teknologi informasi digunakan di dalam pengambilan keputusan bisnis sebuah organisasi. Di dalam suatu organisasi, informasi merupakan sesuatu yang penting didalam mendukung proses pengambilan keputusan oleh pihak manajemen. Menurut Laudon: “Satuan komponen yang saling berhubungan yang mengumpulkan (atau mendapatkan kembali), memproses, menyimpan, dan

mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi”.

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu.

Hubungan antara data dan informasi ditunjukkan pada Gambar 2.4



Gambar 2.4 Hubungan Data dan Informasi

2.3. Expert System

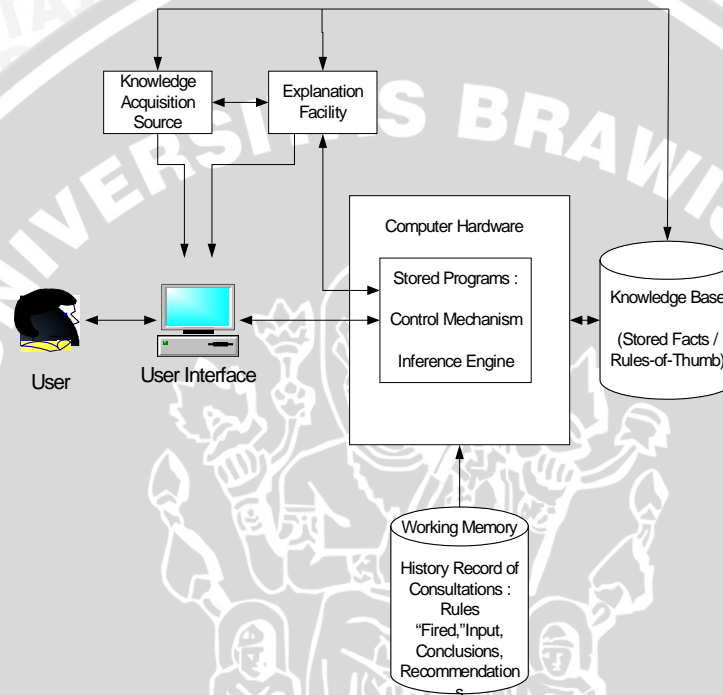
Definisi *Expert System* menurut Durkin (1992:7) adalah sebagai berikut: “*Expert System is a computer program designed to model the problem-solving ability of a human expert.*” Jadi *Expert System* adalah suatu program komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah dari seorang pakar.

Secara umum, *Expert System* adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para pakar. *Expert System* yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para pakar. *Expert System* dapat mengumpulkan dan menyimpan pengetahuan seorang pakar atau beberapa orang pakar ke dalam komputer. Pengetahuan tersebut kemudian digunakan oleh siapa saja yang memerlukannya. Tujuan utama *Expert System* bukan untuk mengganti kedudukan seorang pakar, tetapi hanya untuk memasyarakatkan pengetahuan dan pengalaman para pakar yang keberadaannya cukup jarang. *Expert System* memungkinkan orang lain dapat meningkatkan produktivitasnya, memperbaiki kualitas pengambilan keputusannya, dan memecahkan permasalahan rumit lainnya, sekalipun tanpa kehadiran seorang pakar. Dengan *Expert System* ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan

para ahli. Bagi para ahli sendiri, *Expert System* ini juga akan membantu aktivitasnya sebagai asisten yang berpengalaman.

2.3.1 Elemen *Expert System*

Elemen dari sebuah *Expert System* dapat dijelaskan melalui Gambar 2.5 dibawah ini :



Gambar 2.5. Elemen *Expert System*
Sumber: Dologite, 1992

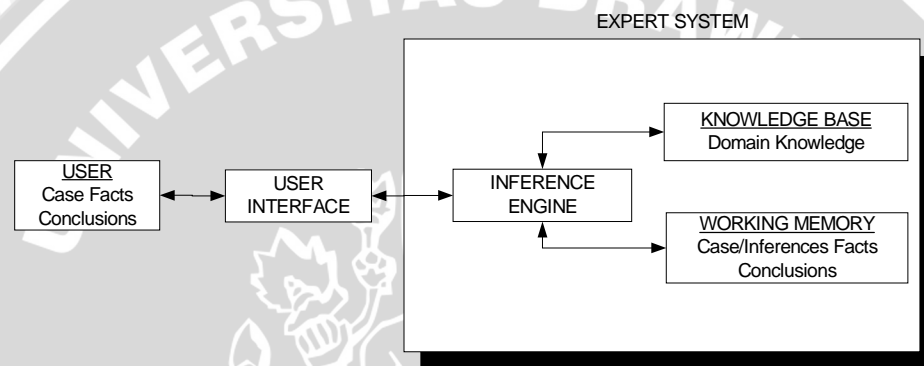
Tiap elemen penyusun *Expert System* dapat dijelaskan sebagai berikut:

- User Interface* adalah penghubung komunikasi antara pemakai (*user*) dengan sistem pakar (*Expert System*).
- Working Memory* adalah database yang digunakan untuk menyimpan fakta yang nantinya digunakan oleh *rules*.
- Explanation Facility* adalah fasilitas yang digunakan untuk menjelaskan sistem pengambilan keputusan kepada pemakai (*user*).
- Inference Engine* sebagai tempat untuk membuat kesimpulan dengan memutuskan *rules* mana yang cocok dengan fakta dan mengeksekusi *rules* yang memiliki prioritas tertinggi.

- e. *Knowledge Acquisition Facility* adalah fasilitas untuk memasukkan pengetahuan kedalam sistem secara otomatis.
- f. *Knowledge Base* adalah fasilitas untuk menyimpan pengetahuan pakar.

2.3.2 Cara Kerja Expert System

Berdasarkan elemen-elemen *Expert System* yang telah dijelaskan melalui gambar 2.5, *Expert System* pada dasarnya terdiri dari tiga komponen utama yaitu : *Inference Engine*, *Knowledge Base*, dan *Working Memory*. Tiga komponen ini dapat digambarkan sebagai berikut :



Gambar 2.6 Diagram Blok Umum *Expert System*
Sumber: Dologite, 1992

2.3.3 Teknik Inferensi

Menurut Luger (1990:94), definisi inferensi adalah: “*Inference is the process used in an expert system of deriving new information from known information*”. Yang dapat diartikan bahwa inferensi adalah proses yang digunakan dalam sebuah *Expert System* untuk menghasilkan informasi baru dari informasi yang telah diketahui.

Mesin inferensi adalah salah satu bagian dari *Expert System* dimana menerapkan teknik inferensi untuk mengolah basis pengetahuan sehingga sistem sampai pada suatu kesimpulan. Teknik inferensi yang biasa digunakan ada dua macam, yaitu:

1. *Forward Chaining*

Forward chaining atau disebut juga pelacakan ke depan adalah suatu metode pelacakan yang dimulai dengan satu atau beberapa fakta awal sehingga didapatkan hasil atau *goal* yang belum diketahui sebelumnya. Proses bergerak maju dengan cara mencocokkan fakta tersebut dengan premis atau bagian *IF* yang ada pada aturan-aturan

dalam basis pengetahuan. Jika saat dieksekusi semua premis bernilai benar (*True*), maka bagian *THEN* akan diambil sebagai fakta baru. Proses pelacakan ini akan dilakukan terus-menerus secara berantai sampai ditemukan hasil atau *goal* yang diinginkan atau sampai tidak terdapat aturan lagi.

Berikut ini adalah contoh dari proses *forward chaining*:

- *Facts* (fakta): A, B, C
- *Rules* (aturan):
 1. *If A & B Then X*
 2. *If X & C Then Y*
 3. *If Y Or D Then Z*
- *Goal?*
- | <u>Langkah</u> | <u>Fakta</u> | <u>Keterangan</u> |
|----------------|--------------|--|
| 0 | ABC | Langkah awal, terdapat 3 fakta. |
| 1 | ABCX | Fakta baru (X) berasal dari aturan 1. |
| 2 | ABCXY | Fakta baru (Y) berasal dari aturan 2. |
| 3 | ABCXYZ | <i>Goal</i> (Z) didapat dari aturan 3. |

2. *Backward Chaining*

Backward chaining atau pelacakan ke belakang adalah suatu metode penalaran yang dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. *Backward chaining* berusaha untuk membuktikan kebenaran suatu hipotesis dengan cara mengumpulkan informasi (fakta-fakta) tambahan.

2.4. Sistem Basis Data

Sistem Basis Data adalah sistem penyimpanan data dengan memakai komputer, dapat juga diartikan sebagai suatu koleksi data yang saling berhubungan yang dapat disusun dalam beberapa cara untuk mendukung pemrosesan dan pencarian pada sistem aplikasi dari suatu organisasi. Menurut Sutedjo (2002), Tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. Elemen-elemen dari basis data adalah sebagai berikut:

1. *Enterprise*: Suatu jenis organisasi.



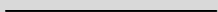

2. *Entity*: Objek pada *enterprise* berdasarkan data yang disimpan.
3. *Attribute (Field)*: Suatu karakter yang terdapat pada *entity*.
4. *Record*: Kumpulan dari beberapa *field*, menunjukkan suatu nilai dari *entity*.
5. *File*: Kumpulan dari beberapa *record* yang berisi data tentang *entity*.

2.4.1 Diagram Entity-Relationship (diagram E-R)

Diagram *entity relationship* merupakan suatu bentuk diagram yang menggambarkan model *entity-relationship* yang berisi komponen-komponen himpunan entitas atau himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari ‘dunia nyata’ yang dicermati (Kroenke, 2005).

Notasi atau simbol yang digunakan dalam diagram E-R (Kroenke, 2005):

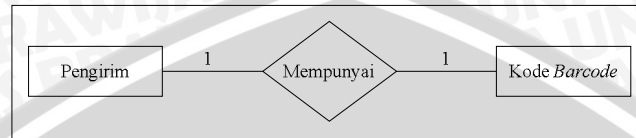
Tabel 2.1 Notasi yang digunakan dalam diagram E-R

Simbol	Definisi
 Entitas	Menunjukkan entitas yang berhubungan dengan sistem
 Atribut	Menunjukkan atribut yang dimiliki oleh entitas
 Link	Menunjukkan <i>link</i>
 Relasi	Menunjukkan relasi antar entitas

Berikut penggambaran relasi antar himpunan entitas, serta relasi itu sendiri digunakan untuk menjelaskan batasan-batasan pada jumlah entitas yang dihubungkan dengan melalui suatu *relationship*. Ada tiga derajat relasi, yaitu (Kroenke, 2005) :

1. *One to one* (satu ke satu)

Dalam relasi satu ke satu, setiap *record* dalam tabel A berhubungan paling banyak dengan satu *record* pada tabel B, dan begitu juga sebaliknya. Jenis relasi ini tidak umum, karena sebenarnya tabel A dan tabel B dapat digabungkan menjadi satu tabel. Contoh: Setiap pengirim mempunyai satu kode *barcode*.



Gambar 2.7 Contoh relasi satu ke satu

2. *One to many* (satu ke banyak)

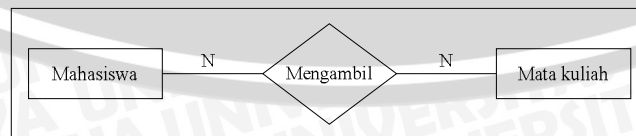
Relasi satu ke banyak adalah bentuk relasi yang paling umum. Dalam relasi satu ke banyak, sebuah *record* dari tabel A berhubungan dengan banyak *record* pada tabel B. Namun sebuah *record* dalam tabel B berhubungan dengan paling banyak dengan satu *record* pada tabel A. Contoh: Setiap pengirim dapat mengirim satu atau lebih barang.



Gambar 2.8 Contoh relasi satu ke banyak

3. *Many to many* (banyak ke banyak)

Dalam relasi banyak ke banyak, sebuah *record* dalam tabel A dapat berhubungan dengan banyak *record* pada tabel B dan sebaliknya. Jenis relasi ini hanya dimungkinkan jika kita mendefinisikan tabel baru sebagai perantara. Contoh: setiap mahasiswa dapat mengambil mata kuliah lebih dari satu dan setiap mata kuliah dapat diambil oleh lebih dari satu mahasiswa.




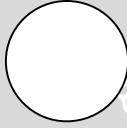
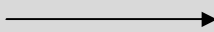
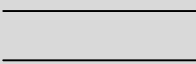
Gambar 2.9 Contoh relasi banyak ke banyak

2.4.2 Data flow Diagram (DFD)

Data flow diagram (DFD) merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yg mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program (Kroenke, 2005).

Adapun simbol-simbol yang sering digunakan dalam *data flow diagram* (DFD) dapat dilihat pada Tabel 2.2:

Tabel 2.2 Simbol-Simbol Pada DFD

Simbol	Definisi
 Entitas Eksternal	Menunjukkan entitas yang berhubungan dengan sistem
 Proses	Menggambarkan proses yang dilakukan oleh sistem
 Objek Data	Menunjukkan arah aliran data
 Penyimpanan Data	Menunjukkan tempat penyimpanan data

Sumber: Kroenke, 2005

Untuk memulai suatu diagram aliran data, diperlukan suatu rangkuman narasi sistem organisasi yang telah dibuat dalam bentuk sebuah daftar dengan empat kategori yang terdiri dari entitas eksternal, aliran data, proses, dan penyimpanan data. Daftar ini digunakan untuk membantu menentukan batas-batas sistem yang akan digambarkan.

Tingkatan-tingkatan yang terdapat pada DFD adalah (Kroenke, 2005) :

1. Diagram konteks

Diagram konteks adalah tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data – aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data – aliran data menuju dan dari sistem diketahui penganalisis dari wawancara dengan pengguna dan sebagai hasil analisis dokumen.

2. Diagram *zero* (diagram 0)

Diagram 0 adalah pengembangan diagram konteks dan bisa mencakup sampai sembilan proses. Memasukkan lebih banyak proses pada level ini akan terjadi dalam suatu diagram yang kacau yang sulit dipahami. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut sebelah kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Penyimpanan data–penyimpanan data utama dari sistem dan semua entitas eksternal dimasukkan kedalam diagram 0.

3. Diagram level n

Diagram level n adalah hasil dekomposisi dari diagram *zero*. Diagram level n menjelaskan proses secara lebih terperinci. Diagram level 1 merupakan turunan langsung dari diagram *zero*, artinya diagram level 1 berada satu tingkat lebih rendah dari diagram *zero*. Apabila diagram level 1 ini diuraikan lagi, maka akan terbentuk diagram level 2, dan seterusnya.

2.5 *Microsoft Access*

2.5.1 Pengenalan *Micosoft Access*

Microsoft Access adalah salah satu program pengelola basis data yang berdaya guna dan fleksibel. *Access* bukanlah suatu program sederhana meskipun *Access* mempunyai beberapa fitur untuk membantu para pemula. *Access* memungkinkan penggunaanya untuk mengumpulkan, menyimpan, dan mengatur informasi seperti halnya membuat laporan yang mengarah kepada kesimpulan akhir.

2.5.2 Penyimpanan Data oleh *Microsoft Access*

Pada *Access*, pertama-tama perlu dibuat satu *file* basis data. *File* tersebut menyimpan segala sesuatu yang dibuat untuk basis data, tidak hanya semua data, tapi juga *form-form*, laporan-laporan, dan indeks.

- **Tabel**
Tabel adalah pusat basis data. Tabel sangat mirip dengan *spreadsheet*. *Access* menyimpan setiap entri basis data pada barisnya sendiri.
- **Form**
Semua data yang dimasukkan ke basis data akan disimpan dalam tabel melalui *form*.
- **Laporan**
Form dirancang untuk dipakai pada layar, sedangkan laporan (*report*) dirancang untuk dicetak. Laporan adalah kumpulan data yang diformat secara khusus dan dikelola menurut spesifikasi *user*.
- **Query**
Query adalah suatu cara untuk membuang informasi yang tidak perlu dilihat sehingga *user* hanya melihat informasi yang diperlukan saja

Berikut ini adalah salah satu tampilan dari *Microsoft Access*:



Gambar 2.10 Gambar Tampilan *Microsoft Access*

2.6 Integrasi *Expert System* dan Sistem Basis Data

Untuk membangun integrasi antara *Expert System* dan basis data, perlu ditentukan terlebih dahulu peran basis data pada suatu *Expert System*. Menurut Paul Harmon dan Brian Sawyer (1990:235), terdapat beberapa cara kerja basis data dalam integrasinya dengan *Expert System*, yaitu:

1. Suatu *Expert System* dapat digunakan sebagai *front end* bagi basis data (*Expert System* ditempatkan di depan basis data). Dalam hal ini, *Expert System* memberikan pertanyaan kepada pengguna pada saat melakukan konsultasi dan

selanjutnya memberikan kesimpulan akhir berdasarkan aturan-aturan yang ada untuk inialisasi *query* basis data.

2. Suatu *Expert System* dapat digunakan sebagai *back end* bagi basis data (*Expert System* ditempatkan di belakang basis data). Dalam hal ini, *Expert System* mengambil hasil *query* dari basis data dan menganalisisnya menggunakan *rule-rule* yang ada untuk kemudian dibuat suatu rekomendasi.
3. Atribut dan nilai dari *rule-rule* pada *Expert System* dapat disimpan pada suatu basis data. Secara umum, cara ini digunakan untuk membuat *Expert System* dengan pendekatan *decision table*.
4. Basis data dapat digunakan untuk menyimpan kasus-kasus yang sebelumnya telah dijalankan pada *Expert System* dengan tujuan untuk mencatat alasan serta rekomendasi yang telah diberikan.

Pada perancangan Skripsi ini, cara integrasi yang digunakan adalah cara yang kedua, yaitu *Expert System* berperan sebagai *front-end* bagi basis data.

2.7 Proses Produksi PT. Yamaha Musical Products Indonesia

Proses produksi merupakan cara atau metode dan teknik untuk menciptakan atau menambah kegunaan suatu barang atau jasa dengan menggunakan sumber-sumber tenaga kerja, mesin, bahan-bahan dan dana yang cukup banyak.

2.7.1 Urutan Proses Produksi Pianika

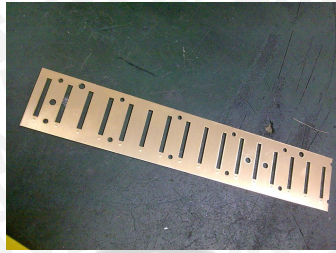
Pembuatan produk pianika secara garis besar digolongkan menjadi tiga tahapan yaitu *reed plate*, *initial process*, dan *final process*.

A. Reed Plate

Proses dalam *reed plate* ini adalah proses produksi *reed* dan *plate* yang merupakan jantung (*heart*) dalam sebuah pianika. Urutan proses produksinya adalah sebagai berikut:

1. Plate

Plate adalah sebuah lempengan yang terbuat dari logam kuningan yang berbentuk persegi panjang dengan komposisi tertentu sesuai dengan tipe pianika.



Gambar 2.11 *Plate*

Sumber: Dokumen KK PT.YMPI, 2008

- 1.1 *Plate Levelling Process* (proses pelurusan plate)
- 1.2 *Plate cutting and Press Process* (proses penekanan & pelubangan plate)
- 1.3 *Plate Washing Process* (proses pencucian plate)

2. *Reed*

Reed adalah sebuah potongan kecil-kecil yang terbuat dari logam kuningan (*Material For Reed*) yang ditempatkan pada *plate*.



Gambar 2.12. *Reed*

Sumber: Dokumen KK PT.YMPI, 2008

- 2.1 *Reed Levelling & Cutting Process* (proses pelurusan & pemotongan reed)
- 2.2 *Reed Shaving Process / Benkuri* (proses penipisan reed)
- 2.3 *Reed Pressing Process / Benuki* (proses pengepresan reed)
- 2.4 *Reed Washing Process* (proses pencucian reed)
3. *Spot Welding Process* (proses pengelasan/penyambungan *plate* dengan *reed*, yang selanjutnya disebut dengan *reedplate*)
4. *Painting* (pengecatan *reedplate*)
5. *Annealing* (proses pemanasan untuk menguatkan *reedplate*)
6. *After Annealing* (pengecekan *reedplate*)

B. Initial Process

1. *Reed Adjustment / Bensuki / Sukimi* (penyetelan celah *reed*)

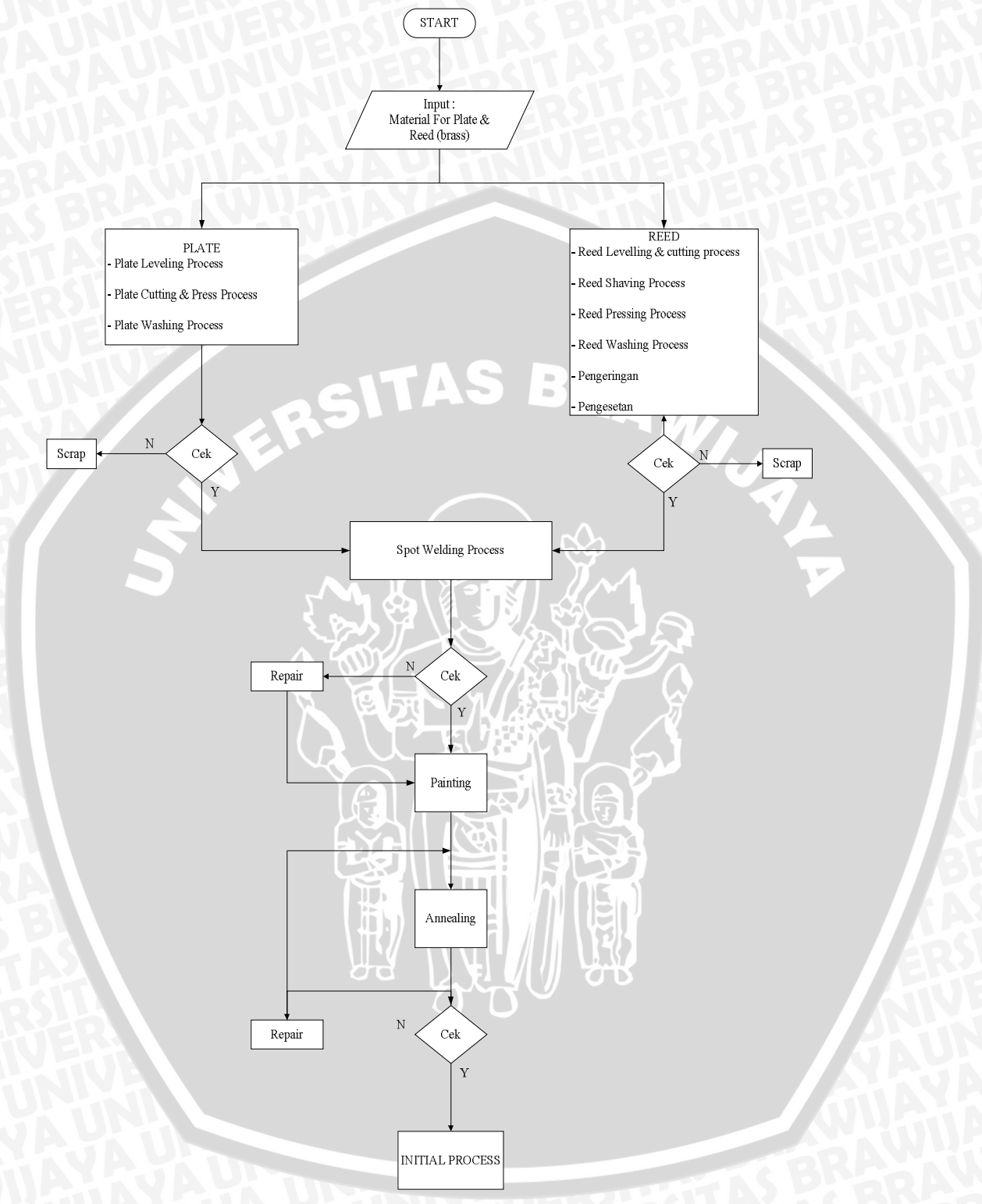
2. *Reed Rising Adjustment / Benage / Agari* (penyetelan pengangkatan reed)
3. *Fixing Plate / Pureto Toritsuke* (pemasangan plate)

C. Final Process

1. *Reed Tuning / Rido Chouritsu* (penyetelan reed)
2. *Rust Prevention / Boseiyu Tofu* (pengolesan minyak anti karat)
3. *Fixing Air Chamber / Kuukisitsu Toritsuke* (pemasangan ruang udara)
4. *Fixing Water Key Valve / Mizunukibarubu Toritsuke* (pemasangan katup pelepas air)
5. *Sound Inspection / Oto kensa* (pemeriksaan suara)
6. *Fixing Biri, Rate / Ritsubiri Shusei* (perbaikan biri, ketepatan nada)
7. *Air Leak Inspection, Numbering / More kensa, Seibanuchi* (pemeriksaan kebocoran udara, dan penomoran)
8. *Fixing Cover Lower / Shita Kaba Toritsuke* (pemasangan tutup bawah)
9. *Fixing Cover Right/Left / Ue Kaba Toritsuke* (pemasangan tutup atas kanan/kiri)
10. *Clean Up / Fukiage* (pengelapan)
11. *Outer View Inspection/ Gaikan Kensa* (pemeriksaan permukaan)
12. *Final Check / Shuseigo Kensa* (pemeriksaan akhir)
13. *Case Storage, Wagon Loading / Kesu Shuunou* (penyimpanan dalam case, peletakan pada wagon)
14. *Packaging/ Konpou* (pengepakan)

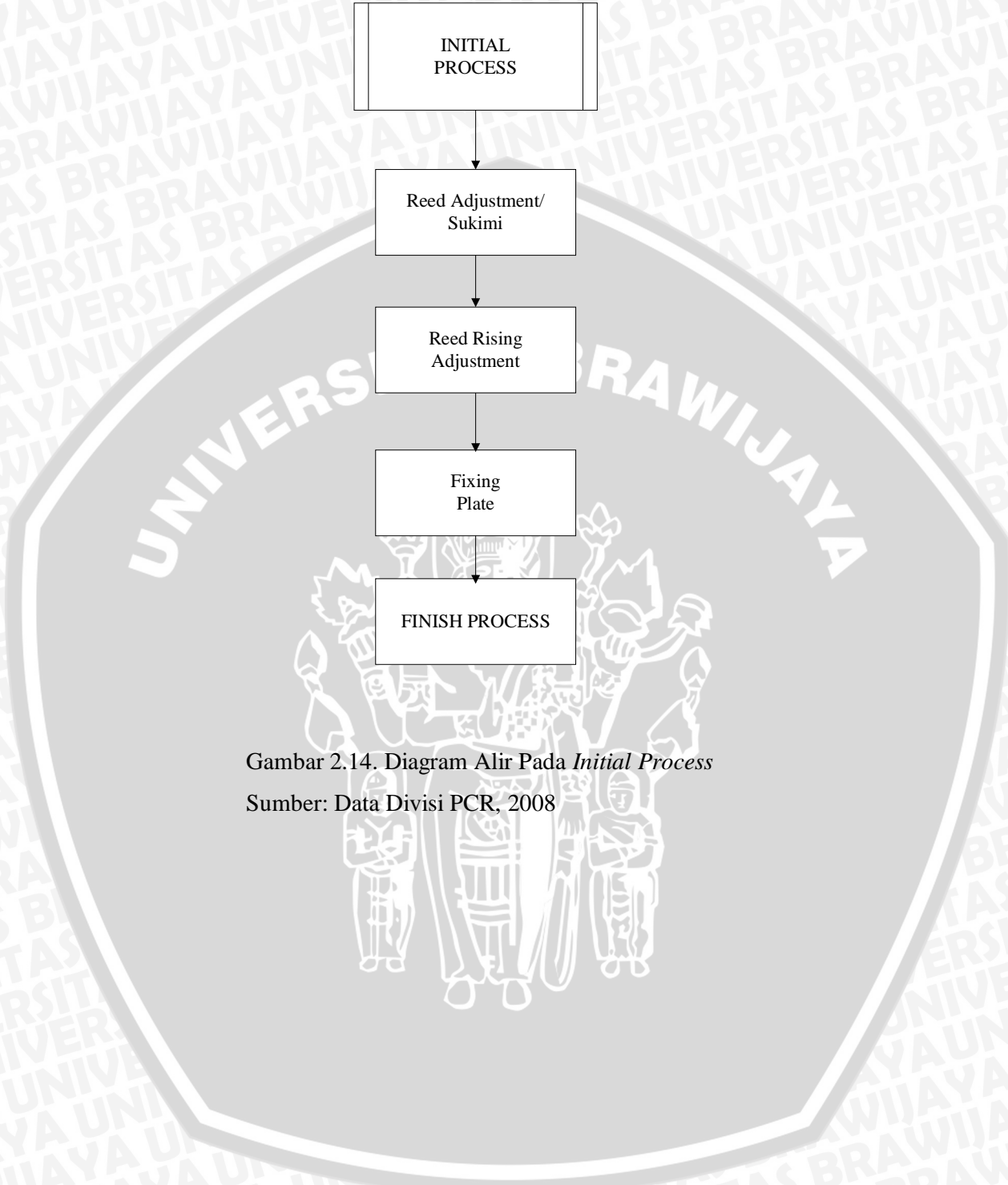
2.7.2 Diagram Alir Proses Produksi Pianika

Pada proses pembuatan pianika terdapat alur proses pembuatannya. Melalui diagram alir dan aliran material maka dapat mempermudah mengetahui proses urutan pembuatan pianika. Diagram alir dan aliran material dapat digambarkan sebagai berikut:

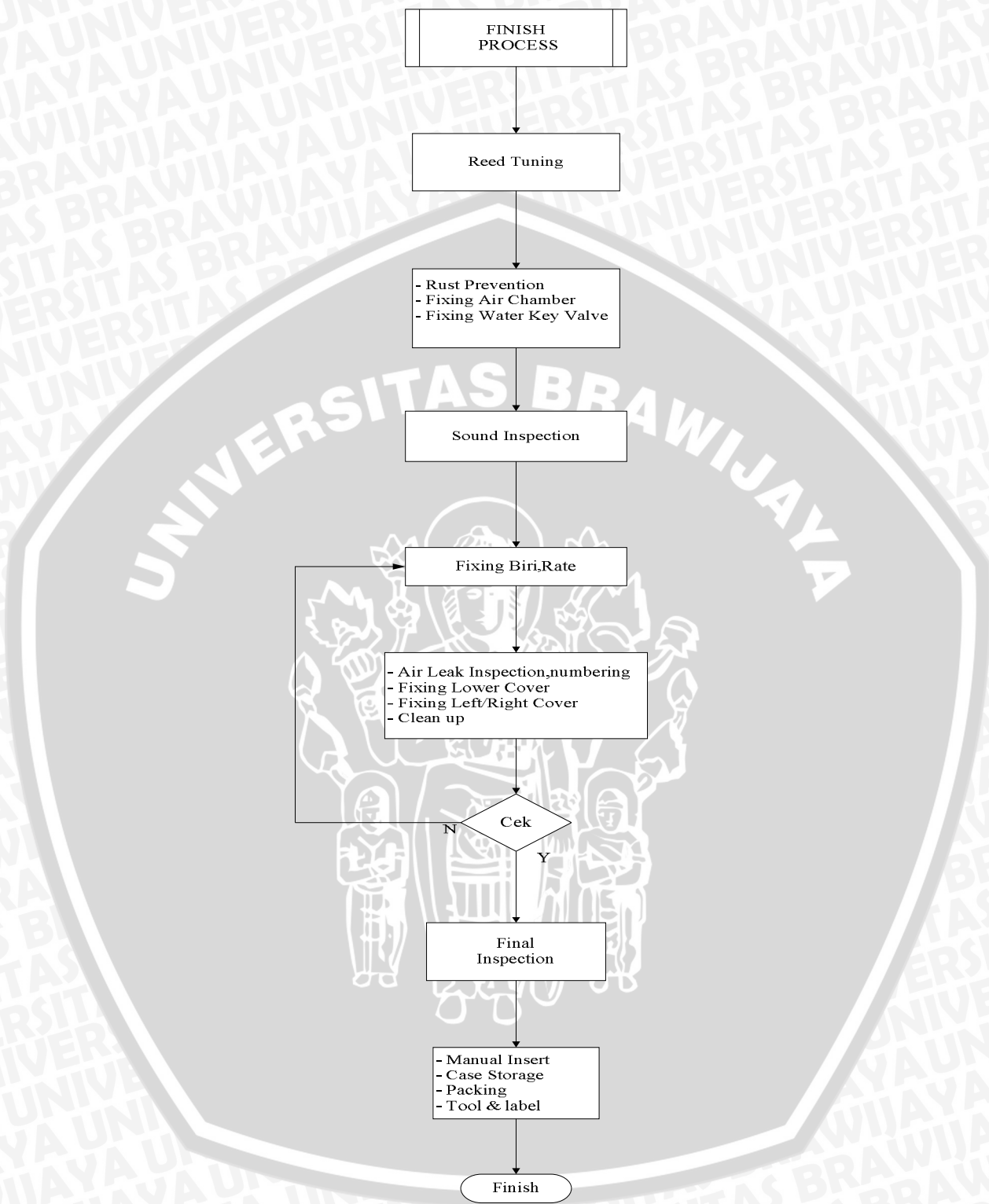


Gambar 2.13. Diagram Alir Pada Proses *Reedplate*

Sumber: Data Divisi PCR, 2008



Gambar 2.14. Diagram Alir Pada *Initial Process*
Sumber: Data Divisi PCR, 2008



Gambar 2.15. Diagram Alir Pada *Finish Process*

Sumber: Data Divisi PCR, 2008

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah :

1. Metode studi literatur yang berhubungan dengan permasalahan yang dibahas dalam penulisan ini baik dari jurnal dan buku di perpustakaan maupun internet.
2. Metode *Descriptive Research*, menggambarkan kondisi perusahaan atau objek sesuai dengan keadaan yang ditemukan atau diamati.

3.2 Penelitian Pendahuluan

Pada langkah ini dilakukan pengamatan terhadap kejadian-kejadian yang terjadi di lapangan. Pengamatan dilakukan di PT. Yamaha Musical Products Indonesia di kota Pasuruan pada bulan Oktober – November tahun 2008. Berdasarkan pengamatan tersebut, ditemukan topik yang menarik untuk dijadikan suatu bahasan dalam skripsi yaitu merancang suatu sistem informasi pengendalian kualitas yang terintegrasi dengan *Expert System* untuk menelusuri penyebab cacat produk.

3.3 Identifikasi Masalah

Setelah dilakukan penelitian pendahuluan pada objek yang telah dipilih, maka tahap selanjutnya adalah mengidentifikasi permasalahan. Dari tahap ini, didapatkan beberapa kendala yang menjadi masalah. Oleh karena itu, perlu dicari solusinya, yaitu dengan merancang sistem informasi pengendalian kualitas terintegrasi dengan *Expert System* untuk membantu menelusuri cacat produk, sehingga didapatkan penyebab serta solusi secara cepat.

3.4 Tujuan Perancangan

Dalam langkah ini ditetapkan tujuan perancangan yang akan digunakan sebagai acuan penilaian dalam menentukan tingkat keberhasilan perancangan yang akan dibuat. Menetapkan tujuan perancangan perlu dilakukan agar perancangan yang akan dibuat menjadi terarah dan terukur tingkat keberhasilannya. Mengingat waktu dan biaya maka

aplikasi yang dirancang adalah hanya *prototype*-nya dengan contoh kasus pengendalian kualitas.

3.5 Studi Literatur

Langkah selanjutnya adalah melakukan studi literatur. Langkah ini dilakukan dengan mengumpulkan dan mempelajari buku, literatur, jurnal, website dan sumber-sumber lain mengenai *Expert System*, Basis Data, serta materi-materi lain yang terkait, seperti : pengendalian kualitas, peta kontrol, dsb.

3.6 Pengumpulan Data

Pengumpulan data adalah pencatatan-pencatatan atau hal-hal atau keterangan-keterangan atau karakteristik-karakteristik sebagian atau seluruh elemen populasi yang akan menunjang atau mendukung penelitian. Adapun teknik pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

- Observasi, merupakan cara pengumpulan data dengan jalan mengamati langsung jalannya aktivitas-aktivitas dari obyek yang diteliti.
- *Interview*, merupakan cara pengumpulan data dengan jalan mengadakan wawancara langsung dengan pihak-pihak di perusahaan yang berkompeten dengan materi penelitian.
- Dokumentasi, merupakan cara pengumpulan data dengan mengambil data-data perusahaan berupa laporan-laporan, catatan-catatan, atau arsip-arsip yang sudah ada.

3.7 Perancangan Sistem

Proses perancangan sistem ini dilakukan sebagai representasi awal suatu program dibuat. Adapun langkah-langkahnya adalah sebagai berikut :

1. Perencanaan (*Planning*)

Pada tahap ini, dilakukan perencanaan program yang sesuai dengan spesifikasi dan kebutuhan yang ingin dicapai agar hasil yang diperoleh dapat optimal.

2. Analisa Sistem (*Conceptual Design*)

Pembuatan desain konsep sekaligus menganalisa apakah sesuai dengan kebutuhan sistem informasi yang terlibat beserta atributnya.

3. Desain Sistem

Desain sistem meliputi 2 bagian:

- a. Perancangan *user interface*
- b. Pembuatan *flowchart* program

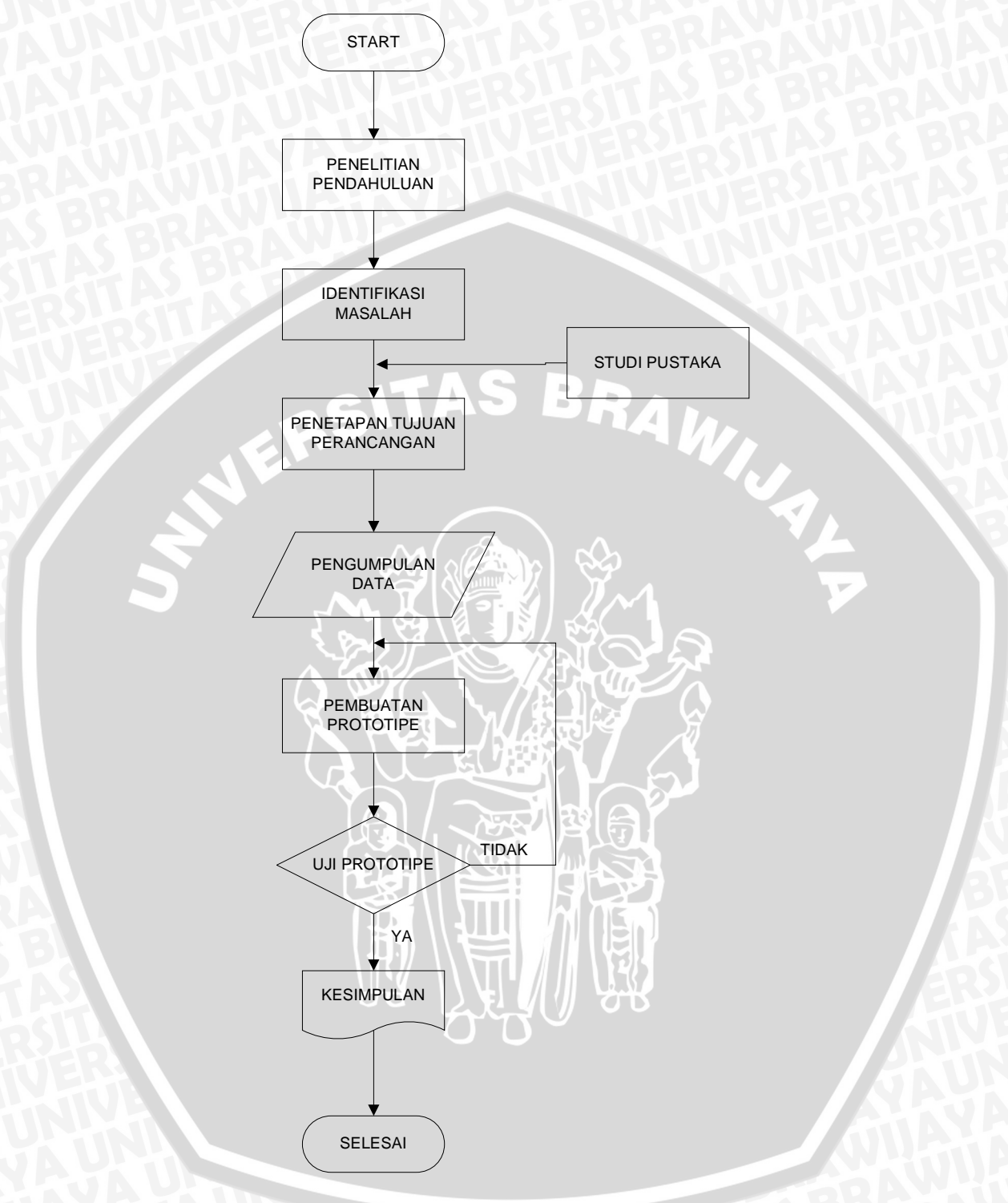
4. Implementasi program

Dalam implementasi program dilakukan langkah-langkah sebagai berikut:

- a. Pembuatan sistem informasi
- b. Pembuatan *expert system*
- c. Pengintegrasian *expert system* dan basis data

3.8 Penarikan Kesimpulan dan Saran

Tahap kesimpulan dan saran merupakan tahap akhir dari metodologi penelitian dan pengembangan. Pada tahap ini, ditarik kesimpulan mengenai apa yang sudah dilakukan dan dicapai dalam pelaksanaan skripsi ini. Kesimpulan haruslah sesuai dengan tujuan yang telah ditetapkan di Bab Pendahuluan. Tahap ini juga memberikan saran-saran yang mungkin dapat dijadikan sebagai masukan bagi yang ingin menindaklanjuti lebih jauh atau mengembangkan *software* aplikasi ini dan juga saran-saran bagi perusahaan yang bersangkutan.



Gambar 3.1. Diagram Alir Metodologi Penelitian



BAB IV

PENGUMPULAN DATA

4.1 Pengantar

Akuisisi Pengetahuan (*Knowledge Acquisition*) adalah proses yang kompleks. Dalam arti yang sempit *Knowledge Acquisition* adalah suatu proses untuk mempelajari, memperoleh, dan mengorganisir pengetahuan dari seorang pakar. Dalam arti yang lebih luas *Knowledge Acquisition* melibatkan semua proses *Knowledge Engineering* termasuk menginterview pakar, menguji dan memperbaiki pengetahuan bersama pakar dan pemakai (*End User*) sampai sistem siap digunakan. Sebagai objek dari pengembangan sistem pakar ini adalah pengendalian kualitas pianika pada PT.YMPI.

4.2 Pengetahuan Pakar (*Expert Knowledge*)

Pengetahuan pakar adalah sumber pengetahuan dari seorang pakar atau beberapa orang pakar sebagai hasil pengalaman kerja bertahun-tahun dalam bidangnya. Pengetahuan ini sering diistilahkan sebagai *Heuristik*. Pengalaman pemecahan masalah ini dapat dijadikan sumber pengetahuan dari sistim pakar yang dibangun.

Untuk penelitian ini, *Expert Knowledge* diperoleh dari beberapa pakar di PT. YMPI dengan pengalaman kerja selama kurang lebih lima tahun. Beberapa contoh *Expert Knowledge* untuk pianika dibahas dibawah ini :

4.2.1 Proses Penyetelan Celah *Reed* (*Reed Adjustment / Bensuki / Sukimi*)

Proses ini adalah proses penyetelan *reed* karena pangkal *reed* kotor, posisi *reed* tidak center terhadap jendela *plate* atau ujung *reed* menyentuh jendela *plate*. Proses ini dilakukan secara manual dengan menggunakan *jig* dengan lampu penerang dan kikir khusus. Sudut pandang operator sangat berpengaruh terhadap hasil produk nantinya.

Metode pengecekan material proses *bensuki* digambarkan sebagai berikut:



Gambar 4.1 Sudut Pandang Cara Melihat *Reed* Pada *Reed Plate* yang Benar (sudut 90°)

Sumber: Dokumen KK PT.YMPI, 2008



Gambar 4.2 Sudut Pandang Cara Melihat *Reed* Pada *Reed Plate* yang Salah (sudut lebih besar/kecil 90°)

Sumber: Dokumen KK PT.YMPI, 2008

Selain sudut pandang operator, terdapat beberapa macam metode yang perlu diperhatikan dalam proses penyetulan celah *reed* ini untuk menghindari adanya *reed yurui* (tidak *center*). Dimulai dari cara memegang *spanner* yang benar, karena dapat berpengaruh terhadap kualitas hasil *Sukimi*. Cara memegang *spanner* dapat digambarkan sebagai berikut:

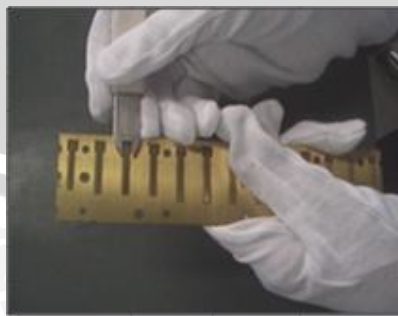


Gambar 4.3 Cara memegang *Spanner* dengan Benar
Sumber: Dokumen KK PT.YMPI, 2008



Gambar 4.4 Cara memegang *Spanner* yang Salah
Sumber: Dokumen KK PT.YMPI, 2008

Penempatan *spanner* pada kepala *reed* tegak lurus dengan arah *reed*. Pastikan *spanner* terpasang pas di kepala *reed*. Gerakan penyetel celah *reed* yang benar adalah dengan cara menggerakkan kepala *reed* ke posisi tengah jendela *plate* dengan menggerakkan ke arah atas atau ke arah bawah. Penempatan *spanner* dan gerakan penyetel celah *reed* dapat digambarkan sebagai berikut:

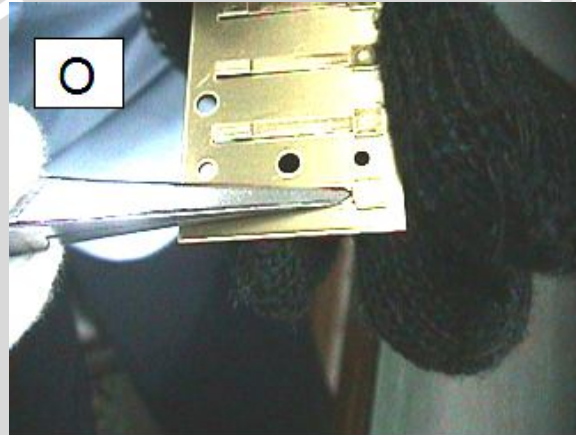


Gambar 4.5 Penempatan *Spanner* Pada Kepala *Reed* yang Benar
Sumber: Dokumen KK PT.YMPI, 2008

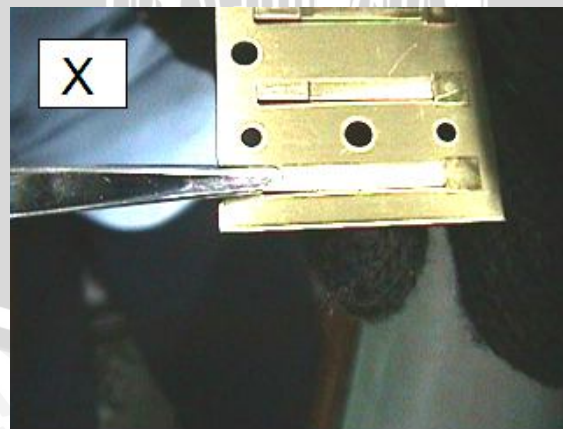
4.2.2 Proses Penyetelan Pengangkatan *Reed* (*Reed Rising Adjustment / Agari*)

Proses ini merupakan proses pembentukan kelengkungan *reed* dari *reed plate*, pada proses ini kelengkungan *reed low* lebih tinggi daripada *reed high*. Proses ini dilakukan secara manual dengan alat Bantu berupa tang jepit yang didesain khusus untuk proses ini.

Pada proses ini, seringkali ditemukan *reed* yang terpasang miring sehingga perlu diluruskan sesuai dengan ketentuan. Operator *Agari* seringkali menggunakan metode perbaikan *reed* yang salah, yaitu dengan meluruskan dan menariknya dari ujung *reed*. Metode perbaikan *reed* yang benar adalah dengan cara meluruskan dan menariknya dari pangkal *reed*. Metode perbaikan *reed* miring dari pangkal dapa digambarkan sebagai berikut:



Gambar 4.6 Cara Perbaikan *Reed* yang Benar
Sumber: Dokumen KK PT.YMPI, 2008

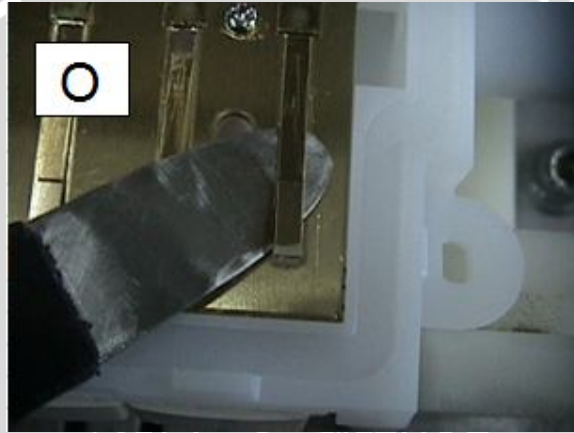


Gambar 4.7 Cara Perbaikan *Reed* yang Salah
Sumber: Dokumen KK PT.YMPI, 2008

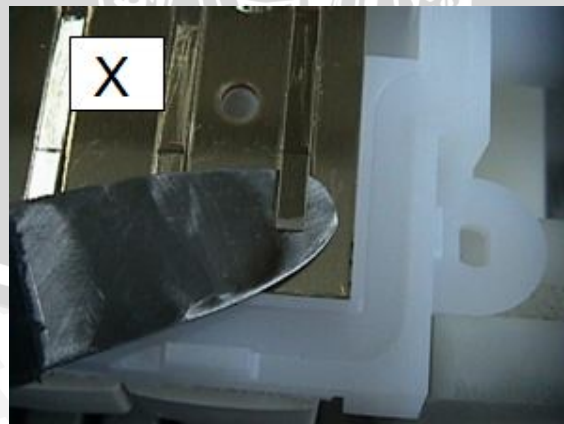
4.2.3 Proses Penyetelan *Reed* (*Reed Tuning / Rido Chouritsu*)

Proses ini adalah proses penyetelan ketepatan *tuning*. Nada tiap *reed* dari *reed plate* disetel dengan cara pengikiran pada tiap-tiap *reed*. Proses ini menggunakan alat Bantu mesin berupa mesin *tuning* dan kikir khusus.

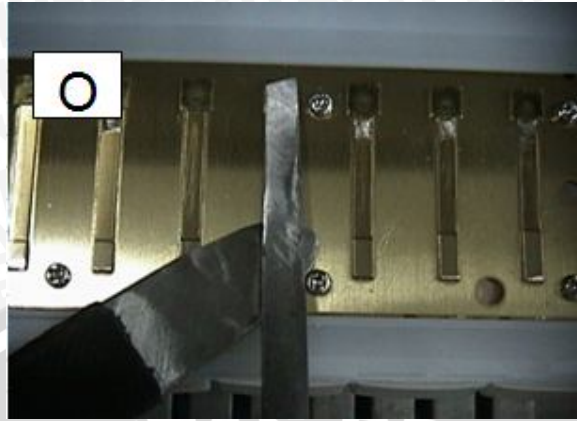
Perlu diketahui pada proses ini merupakan proses yang sangat menentukan dalam ketepatan nada pianika. Posisi hera (landasan reed pada saat pengikiran) juga sangat menentukan agar hasil proses sebelumnya (Agari) tetap, karena tekanan kikiran. Posisi Hera yang benar adalah $\frac{2}{3}$ panjang reed. Sedangkan yang biasa dijumpai adalah posisi Hera terlalu ke ujung reed. Selain itu, yang perlu diperhatikan pada saat proses pengikiran adalah posisi pengikiran dengan arah kikiran lurus, agar hasil kikiran lurus/rata.



Gambar 4.8 Posisi Hera yang Benar
Sumber: Dokumen KK PT.YMPI, 2008



Gambar 4.9 Posisi Hera yang Salah
Sumber: Dokumen KK PT.YMPI, 2008



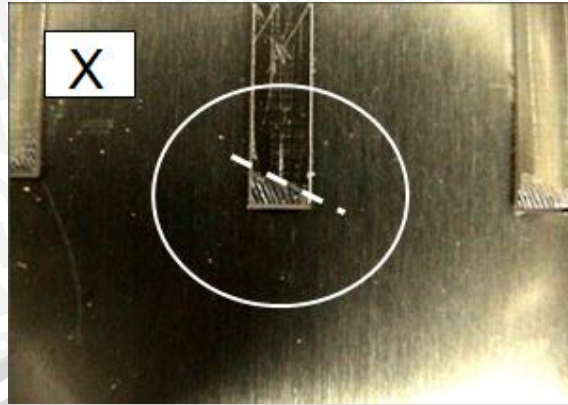
Gambar 4.10 Posisi Pengikiran dengan Arah Kikir Lurus
Sumber: Dokumen KK PT.YMPI, 2008



Gambar 4.11 Posisi Pengikiran dengan Arah Kikir Miring
Sumber: Dokumen KK PT.YMPI, 2008



Gambar 4.12 Hasil Pengikiran yang Lurus / Rata
Sumber: Dokumen KK PT.YMPI, 2008



Gambar 4.13 Hasil Pengikiran yang Miring
Sumber: Dokumen KK PT.YMPI, 2008

4.3 Data Audit Produk

Data audit produk diperoleh dari penelitian pendahuluan yang telah dilakukan. Data diambil dari proses *final check* / *Shuseigo Kensha* (pemeriksaan akhir) selama 3 bulan.



Tabel 4.1 Audit *Finished Good* P32 Juli 2008

No	Tanggal	n	p	Jenis NG Finish Good untuk variabel Bunyi					
				1	2	3	4	5	6
1	1	2120	22	6	7	5	4	0	0
2	2	2120	19	5	7	4	3	0	0
3	3	2120	20	5	6	5	4	0	0
4	4	2120	19	5	7	3	4	0	0
5	5	2120	21	5	4	6	6	0	0
6	7	2120	22	4	6	8	4	0	0
7	8	2120	24	6	8	6	4	0	0
8	9	2120	22	6	5	6	5	0	0
9	10	2120	21	7	6	6	2	0	0
10	11	2120	19	6	4	4	5	0	0
11	12	2120	22	7	8	4	3	0	0
12	14	2120	21	4	9	5	3	0	0
13	15	2120	18	5	8	3	2	0	0
14	16	2120	16	4	9	3	0	0	0
15	17	2120	20	5	6	6	3	0	0
16	18	2120	22	5	6	8	3	0	0
17	22	2120	21	7	5	6	3	0	0
18	21	2120	19	4	8	3	4	0	0
19	24	2120	22	4	11	4	3	0	0
20	25	2120	20	6	4	7	3	0	0
21	26	2120	22	3	7	8	4	0	0
22	28	2120	21	7	5	5	4	0	0
23	29	2120	22	6	7	7	2	0	0
24	31	2120	22	6	5	6	5	0	0
	Jumlah	50880	497	129	160	131	83	0	0
	Rata-rata		20.708	5.3333	6.5833	5.3333	3.4583	0	0

Sumber: Pengolah data

Tabel 4.2 Audit *Finished Good* P32 Agustus 2008

No	Tanggal	n	p	Jenis NG Finish Good untuk variabel Bunyi					
				1	2	3	4	5	6
1	1	2080	19	5	4	5	5	0	0
2	4	2080	20	4	9	5	2	0	0
3	5	2080	17	5	5	5	2	0	0
4	6	2080	18	3	9	4	2	0	0
5	7	2080	20	8	5	7	0	0	0
6	8	2080	14	2	5	5	2	0	0
7	11	2080	15	4	5	4	2	0	0
8	12	2080	15	1	8	5	1	0	0
9	13	2080	17	4	4	7	2	0	0
10	14	2080	22	4	7	7	4	0	0
11	19	2080	19	5	4	6	4	0	0
12	20	2080	19	6	7	5	1	0	0
13	21	2080	17	2	8	3	4	0	0
14	22	2080	17	2	8	4	3	0	0
15	24	2080	15	3	7	2	3	0	0
16	25	2080	16	3	6	3	4	0	0
17	26	2080	18	8	6	3	1	0	0
18	27	2080	16	5	6	2	3	0	0
19	28	2080	14	2	7	1	4	0	0
20	29	2080	18	2	6	5	5	0	0
21	30	2080	16	3	6	4	3	0	0
	Jumlah	43680	362	81	132	92	57	0	0
	Rata-rata		17.238	3.8571	6.2857	4.381	2.7143	0	0

Sumber: Pengolah data

Tabel 4.3 Audit *Finished Good* P32 September 2008

No	Tanggal	n	p	Jenis NG Finish Good untuk variabel Bunyi					
				1	2	3	4	5	6
1	3	2200	20	6	8	3	3	0	0
2	4	2200	22	6	6	6	4	0	0
3	5	2200	17	5	3	2	7	0	0
4	7	2200	17	6	4	3	4	0	0
5	8	2200	18	3	7	4	4	0	0
6	9	2200	23	7	4	9	3	0	0
7	10	2200	16	2	8	3	3	0	0
8	11	2200	19	6	6	3	4	0	0
9	12	2200	16	2	3	5	6	0	0
10	13	2200	20	6	7	4	3	0	0
11	15	2200	17	1	8	3	2	1	2
12	16	2200	22	8	6	4	4	0	0
13	17	2200	23	5	7	8	3	0	0
14	19	2200	22	8	4	6	4	0	0
15	21	2200	17	4	4	5	4	0	0
16	22	2200	21	3	5	9	4	0	0
17	23	2200	22	6	5	5	6	0	0
18	24	2200	18	3	5	4	6	0	0
19	25	2200	21	2	4	4	11	0	0
20	26	2200	20	3	7	5	5	0	0
21	27	2200	24	5	8	6	5	0	0
	Jumlah	46200	415	97	119	101	95	1	2
	Rata-rata		19.762	4.619	5.6667	4.8095	4.5238	0.0476	0.0952

Sumber: Pengolah data

Keterangan tabel:

n = Total produksi aktual (*pcs*)

p = Total cacat produk (*pcs*)

1 = oktaf aneh

2 = Biri

3 = *Reed* tinggi

4 = *Reed* rendah

5 = Bunyi tersumbat

6 = Tidak bunyi

BAB V

PERANCANGAN DAN PENGUJIAN *SOFTWARE* APLIKASI

5.1 Pengantar

Perancangan sistem adalah tahap yang berguna untuk memperbaiki efisiensi kerja suatu sistem yang sudah ada. Tahap perancangan sistem dapat digambarkan sebagai tahap untuk membangun suatu sistem dan mengkonfigurasi komponen-komponen perangkat lunak dan perangkat kerasnya, sehingga menghasilkan sistem yang lebih baik. Aplikasi ini diberi nama Sistem Informasi pengendalian Kualitas dan *Expert System* (SIPKES).

5.2 Perencanaan

Data-data yang telah dikumpulkan digunakan dalam tahap ini untuk menentukan spesifikasi SIPKES secara rinci. Data-data tersebut, antara lain data jumlah produksi aktual (harian), jumlah cacat produk, data pegawai, data penyebab cacat produk dan data solusi cacat produk. Produk yang menjadi bahan penelitian adalah pianika dengan 3 macam jenis, antara lain P25, P32, dan P37.

5.2.1 Spesifikasi *Software*

Atas dasar data-data yang telah terkumpul, dapat dijabarkan spesifikasi aplikasi yang sesuai dengan kebutuhan.

Spesifikasi aplikasi meliputi beberapa hal, antara lain:

1. *File-file* yang dikelola oleh *database* meliputi data cacat produk, data pegawai, data penyebab cacat produk dari 4 faktor, antara lain: mesin/alat, metode, manusia, dan material.
2. *Form-form* yang digunakan dalam aplikasi antara lain: *Form* Input Data Audit (P25, P32, dan P37), *Form* SQC, *Form* Peta Kontrol, *Form* Diagram Batang, *Form* Pareto Diagram, *Form* Pegawai, *Form* Buat Laporan, *Form* Input *Expert System*, *Form* Decision, *Form* Master Data, *Form* Solusi dan *Form* Switchboard. Selain itu juga terdapat *form* tambahan atau sebagai pendukung aplikasi saja, yaitu: *startup screen* dan *pop up expert system*.

3. Laporan-laporan yang digunakan dalam aplikasi antara lain: laporan cacat produk bulanan, laporan cacat produk tahunan, laporan pegawai, laporan hasil *Expert System*.
4. *Statistical Quality Control* yang dipakai adalah diagram batang, Pareto diagram, peta kontrol np, dan *fishbone* diagram.

5.2.2 List Entity

Aplikasi ini dirancang dengan menggunakan konsep sistem Manajemen Basis Data (*Database Management System*). DBMS adalah kumpulan *file* yang saling berkaitan bersama dengan program untuk pengelolannya. *Entity* dan atribut penyusun aplikasi dapat dilihat pada Tabel 5.1. *Entity-entity* tersebut dapat berkembang seiring dengan tahapan analisis sistem selanjutnya.

Tabel 5.1 List Entity

no	Entity	Atribut
1	P25data	ID data Nama Tanggal Produksi aktual Oktaf aneh Biri Reed tinggi Reed rendah Bunyi Tersumbat Tidak bunyi

Tabel 5.1 List Entity (Lanjutan)

no	Entity	Atribut
2	P32 data	ID data Nama Tanggal Produksi aktual Oktaf aneh Biri Reed tinggi Reed rendah Bunyi Tersumbat Tidak bunyi
3	P37data	ID data Nama Tanggal Produksi aktual Oktaf aneh Biri Reed tinggi Reed rendah Bunyi Tersumbat Tidak bunyi
4	Pegawai	NIK Nama Job tittle Tempat lahir Tanggal lahir Alamat Telepon

Tabel 5.1 List Entity (Lanjutan)

no	Entity	Atribut
5	Input Expert System	No Input Nama Tanggal Identifikasi Awal Kerusakan mesin Kesalahan metode Kesalahan operator Kondisi material
6	Identifikasi Awal	Kode identifikasi awal Deskripsi identifikasi
7	Kerusakan Mesin	Kode kerusakan mesin Deskripsi Kerusakan
8	Kesalahan Metode	Kode kesalahan metode Deskripsi kesalahan
9	Kesalahan Operator	Kode kesalahan operator Deskripsi kesalahan
10	Kondisi Material	Kode kondisi material Deskripsi kondisi material
11	Proses kritis	Kode proses Nama proses
12	Solusi	Kode solusi Deskripsi solusi
13	Defect	Kode Defect Deskripsi Defect

Tabel 5.1 List Entity (Lanjutan)

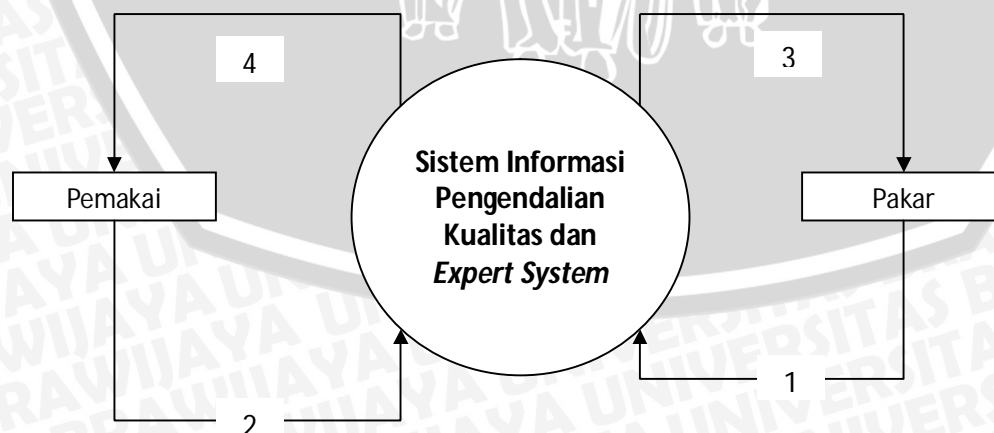
no	Entity	Atribut
14	Decision	Kode Aturan Identifikasi awal Kerusakan mesin Kesalahan metode Kesalahan operator Kondisi material Proses kritis Solusi

5.3 Analisa Sistem

Analisa sistem memiliki tujuan untuk membuat desain konsep sekaligus menganalisa apakah sesuai dengan kebutuhan sistem informasi yang terlibat beserta atributnya. Yang termasuk dalam analisa adalah studi tentang situasi yang dihadapi sistem saat ini, studi tentang sistem informasi yang berkaitan dengan permasalahan yang dihadapi, termasuk menentukan apa yang dibutuhkan oleh sistem tersebut.

5.3.1 Diagram konteks

Langkah pertama dalam analisa sistem adalah membuat konteks diagram. Konteks diagram menekankan pada interaksi sistem dengan lingkungan yang akan mengaksesnya, sehingga sistem dianggap sebagai satu objek dan tidak dijelaskan secara rinci. Konteks diagram SIPKES dapat dilihat pada Gambar 5.1.



Gambar 5.1 Diagram Konteks Aplikasi

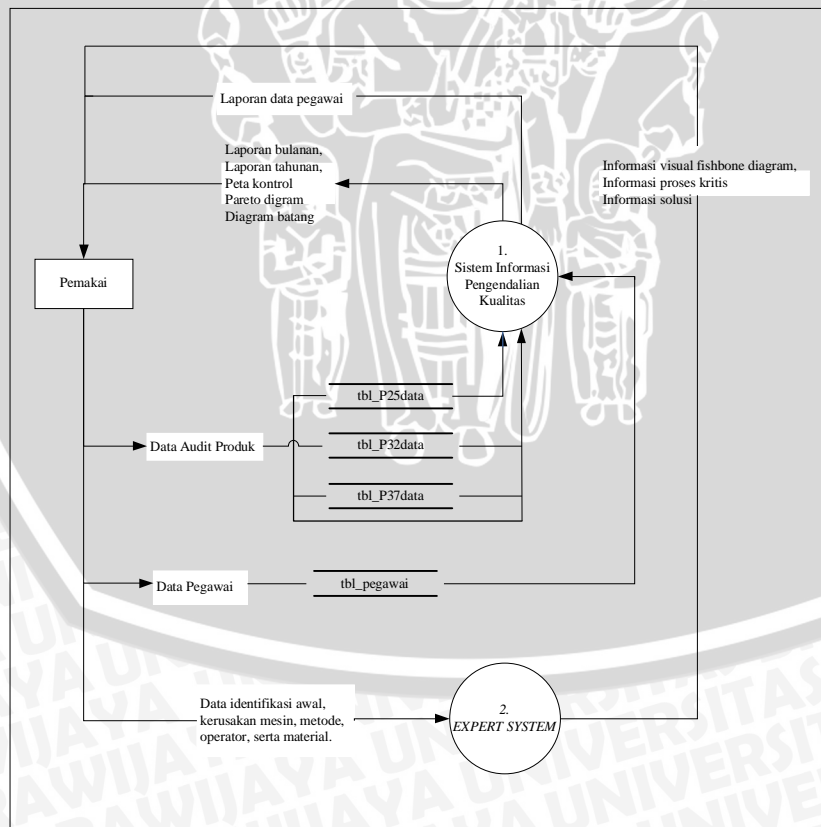
Keterangan dari Gambar 5.1 :

1.
 - a. Kelola data pakar
 - b. Kelola basis pengetahuan
 - c. Kelola basis aturan
2.
 - a. Penelusuran kecacatan produk
 - b. Akses data cacat produk
3.
 - a. Daftar basis pengetahuan
 - b. Daftar basis aturan
4.
 - a. Informasi hasil diagnosa (solusi)
 - b. Informasi visual *fishbone* diagram
 - c. Informasi visual peta kontrol
 - d. Informasi visual pareto diagram
 - d. Informasi data cacat tiap bulan
 - e. Informasi data cacat tiap tahun

5.3.2 Data Flow Diagram (DFD)

DFD adalah model yang digunakan untuk menggambarkan aliran peristiwa yang terjadi dalam sistem. DFD ini nantinya akan menjadi acuan untuk pembuatan *software*.

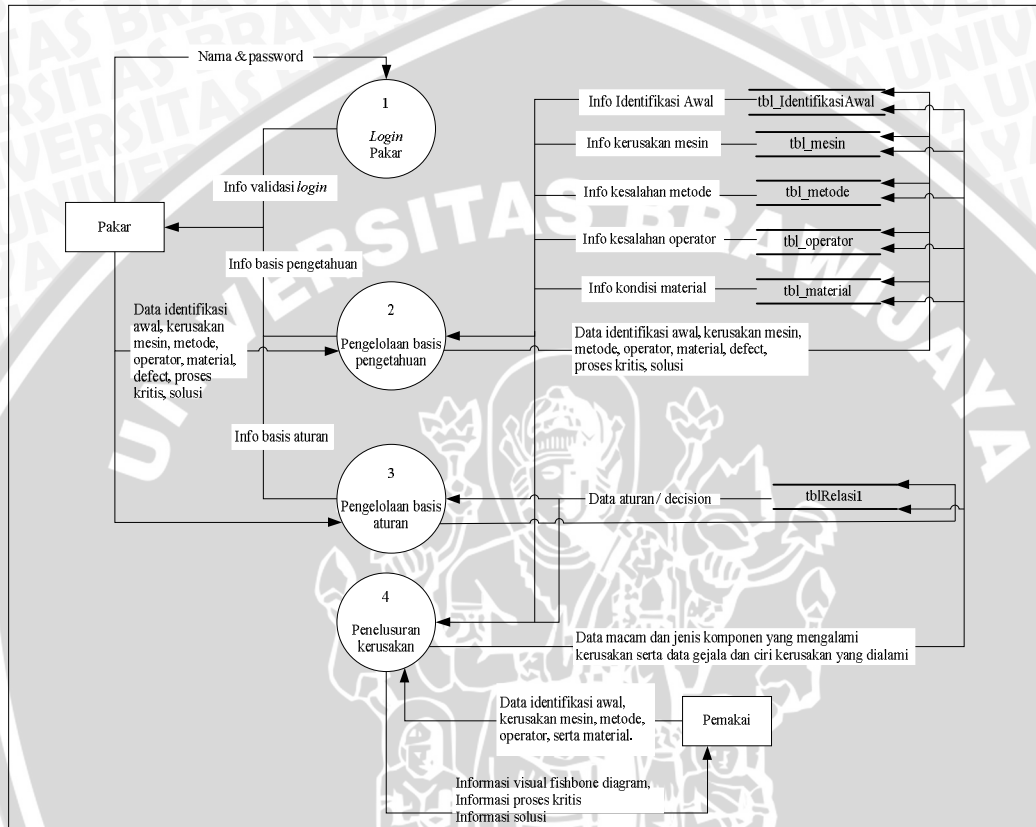
DFD level 0 untuk SIPKES dapat dilihat pada Gambar 5.2.



Gambar 5.2 DFD Level 0 SIPKES

5.3.3 DFD level 1 Proses Expert System

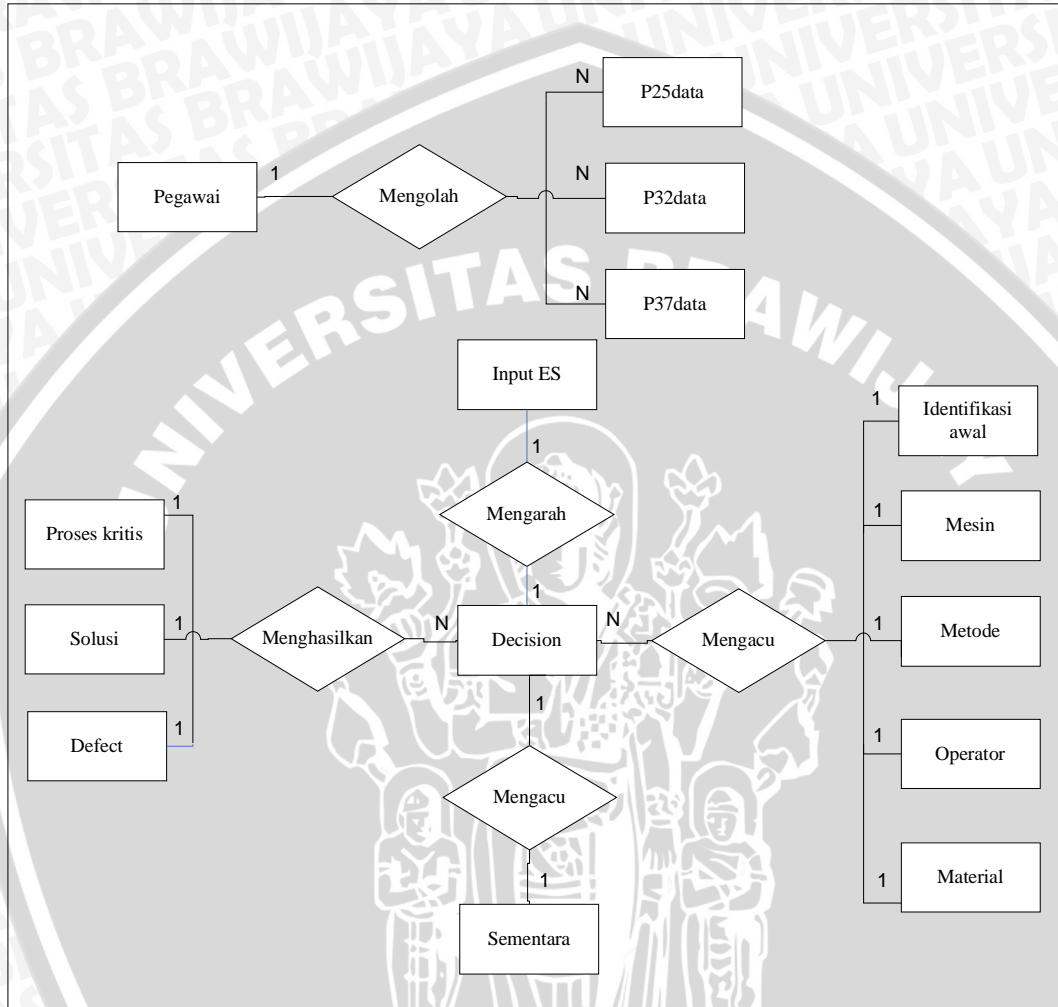
Dalam proses ini ada 4 subproses antara lain: login admin, Pengelolaan basis pengetahuan, pengelolaan basis aturan, penelusuran cacat produk. Subproses ini aliran datanya dapat digambarkan pada Gambar 5.3.



Gambar 5.3 DFD Level 1 Proses *Expert System*

5.3.4 Entity Relationship Diagram (ERD)

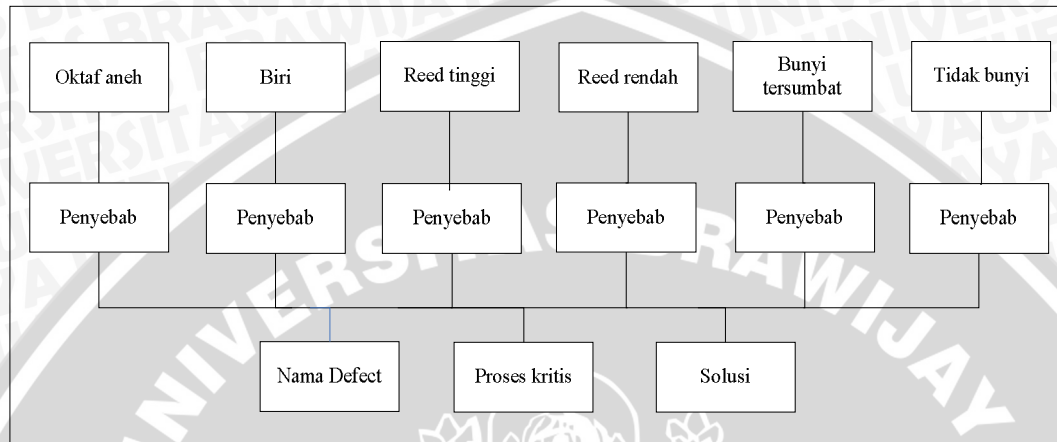
ERD menggambarkan hubungan antar *entity* yang ada dan merupakan konsep awal perancangan tabel yang menjadi dasar dari basis data. Dari tabel-tabel tersebut dapat dibuat ERD seperti Gambar 5.4.



Gambar 5.4 Entity Relationship Diagram SIPKES

5.3.5 Perancangan Blok Diagram

Perancangan blok diagram menunjukkan alur penelusuran *expert system* terhadap parameter-parameter kualitas produk pianika. Blok diagram target keputusan ditunjukkan pada Gambar 5.5.

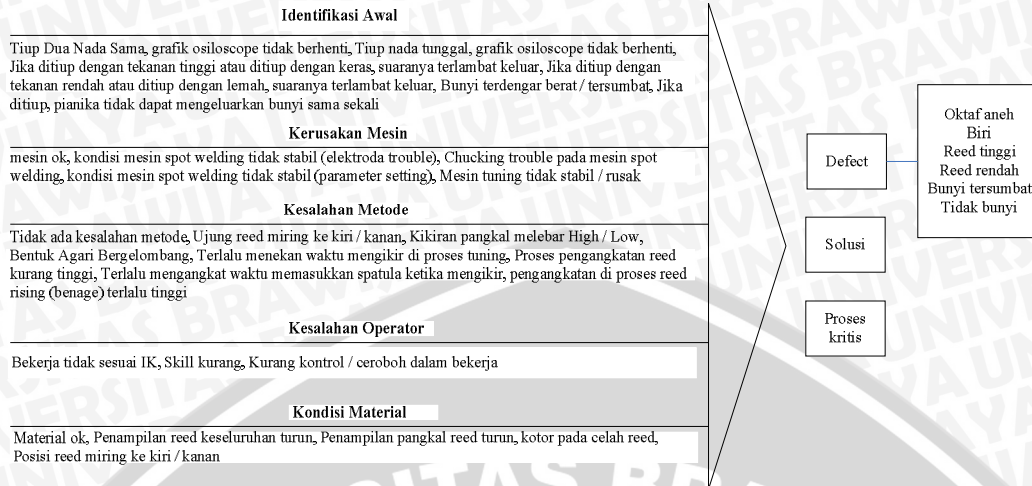


Gambar 5.5 Blok Diagram *Expert System*

Diagram tersebut menunjukkan bahwa sistem akan memberikan informasi nama cacat yang terjadi, solusi tentang tindakan-tindakan perbaikan yang harus dilakukan untuk mengatasi permasalahan mengenai cacat produk pada pianika sekaligus memberikan informasi proses mana yang perlu perbaikan. Rekomendasi yang diberikan oleh *expert system* merupakan hasil kesimpulan dari data-data yang diberikan oleh *user*.

5.3.6 Perancangan *Dependency Diagram*

Dependency diagram adalah sebuah diagram yang mengindikasikan hubungan atau ketergantungan antara identifikasi awal, penyebab cacat produk dari 4 faktor (mesin, metode, operator, dan material), proses kritis, cacat yang terjadi dan solusi yang dibuat oleh prototipe *expert system*. Terdapat beberapa simbol gambar dari *dependency diagram*, yaitu segitiga, persegi panjang dan simbol garis. Simbol segitiga menunjukkan proses diagnosa yang dilakukan oleh sistem pakar, yaitu dengan mengeksekusi rule yang dibentuk. Simbol persegi panjang merupakan simbol yang menggambarkan hasil kesimpulan eksekusi dari suatu rule. Sedangkan simbol garis menunjukkan alur penelusuran *expert system*. *Dependency diagram* proses *expert system*, dapat digambarkan pada Gambar 5.6.



Gambar 5.6 *Dependency Diagram* Proses *Expert System*

Dari gambar tersebut, dapat dilihat bahwa alur penelusuran yaitu: kombinasi identifikasi awal, kerusakan mesin, kesalahan metode, kesalahan operator, dan kondisi material. Kemudian sistem akan melakukan penelusuran dan akan menghasilkan solusi, proses kritis dan cacat yang terjadi (*defect*) sesuai dengan data kombinasi kelima faktor tersebut.

5.4 Desain Sistem

Tahap desain sistem memiliki tujuan untuk mengubah model informasi yang telah dibuat selama tahapan analisis menjadi model yang sesuai dengan teknologi yang akan dipergunakan untuk implementasi sistem informasi.

5.4.1 Perancangan Database Fisik

Tujuan utama tahap ini adalah untuk mengimplementasikan database sebagai suatu himpunan *record*, *file*, *indeks*, atau struktur data lainnya. SIPKES mempunyai 14 tabel yang menjadi tabel utama. Rincian tabel-tabel adalah sebagai berikut:

Tabel 5.2. Tabel P25data

Field	Type Data	Panjang	Keterangan	Key
ID_data	AutoNumber	-	No data	PK
Nama	Text	30	Nama pegawai/operator	
Tanggal	Date/Time	-	Tanggal audit	
n	Number	-	Produksi aktual	
Oktaf_aneh	Number	-	Jumlah cacat	
Biri	Number	-	Jumlah cacat	
Reed_tinggi	Number	-	Jumlah cacat	
Reed_rendah	Number	-	Jumlah cacat	
Bunyi_tersumbat	Number	-	Jumlah cacat	
Tidak_bunyi	Number	-	Jumlah cacat	

Tabel 5.3. Tabel P32data

Field	Type Data	Panjang	Keterangan	Key
ID_data	AutoNumber	-	No data	PK
Nama	Text	30	Nama pegawai/operator	
Tanggal	Date/Time	-	Tanggal audit	
n	Number	-	Produksi aktual	
Oktaf_aneh	Number	-	Jumlah cacat	
Biri	Number	-	Jumlah cacat	
Reed_tinggi	Number	-	Jumlah cacat	
Reed_rendah	Number	-	Jumlah cacat	
Bunyi_tersumbat	Number	-	Jumlah cacat	
Tidak_bunyi	Number	-	Jumlah cacat	

Tabel 5.4. Tabel P37data

Field	Type Data	Panjang	Keterangan	Key
ID_data	AutoNumber	-	No data	PK
Nama	Text	30	Nama pegawai/operator	
Tanggal	Date/Time	-	Tanggal audit	
n	Number	-	Produksi aktual	
Oktaf_aneh	Number	-	Jumlah cacat	
Biri	Number	-	Jumlah cacat	
Reed_tinggi	Number	-	Jumlah cacat	
Reed_rendah	Number	-	Jumlah cacat	
Bunyi_tersumbat	Number	-	Jumlah cacat	
Tidak_bunyi	Number	-	Jumlah cacat	

Tabel 5.5. Tabel Pegawai

Field	Type Data	Panjang	Keterangan	Key
NIK	Text	10	No Induk	PK
Nama	Text	30	Nama pegawai/operator	
Job Title	Text	50	Jabatan	
Tempat lahir	Text	20	Tempat lahir	
Tanggal lahir	Date/Time	-	Tanggal lahir	
Alamat	Text	100	Alamat	
Telepon	Number	-	Telepon	
Keterangan	Memo	-	Keterangan	

Tabel 5.6. Tabel Input Expert System

Field	Type Data	Panjang	Keterangan	Key
ID_input	AutoNumber	-	No input	PK
Nama	Text	30	Nama pemakai/user	
Tanggal	Date/Time	-	Tanggal penggunaan ES	
P1	Text	-	Input identifikasi awal	
P2	Text	-	Input kerusakan mesin	
P3	Text	-	Input kesalahan metode	
P4	Text	-	Input kesalahan operator	
P5	Text	-	Input kondisi material	

Tabel 5.7. Tabel Identifikasi Awal

Field	Type Data	Panjang	Keterangan	Key
ID_Identifikasi	Text	4	No identifikasi awal	PK
IdentifikasiAwal	Text	-	Deskripsi identifikasi	

Tabel 5.8. Tabel Mesin

Field	Type Data	Panjang	Keterangan	Key
IDMesin	Text	4	No kerusakan	PK
CiriKerusakanMesin	Text	-	Deskripsi kerusakan	

Tabel 5.9. Tabel Metode

Field	Type Data	Panjang	Keterangan	Key
IDMetode	Text	4	No kesalahan metode	PK
CiriMetode	Text	-	Deskripsi kesalahan metode	

Tabel 5.10. Tabel Operator

Field	Type Data	Panjang	Keterangan	Key
IDOperator	Text	4	No kesalahan operator	PK
CiriOperator	Text	-	Deskripsi kesalahan operator	

Tabel 5.11. Tabel Material

Field	Type Data	Panjang	Keterangan	Key
IDMaterial	Text	4	No kondisi material	PK
CiriMaterial	Text	-	Deskripsi kondisi material	

Tabel 5.12. Tabel Proses

Field	Type Data	Panjang	Keterangan	Key
IDProses	Text	4	No proses kritis	PK
NamaProses	Text	-	Nama proses kritis	

Tabel 5.13. Tabel Solusi

Field	Type Data	Panjang	Keterangan	Key
ID_solusi	Text	4	No solusi	PK
NamaSolusi	Memo	-	Deskripsi solusi	

Tabel 5.14. Tabel Defect

Field	Type Data	Panjang	Keterangan	Key
IDDefect	Text	4	No defect / cacat	PK
NamaDefect	Text	-	Nama defect / cacat	

Tabel 5.15. Tabel Decision

Field	Type Data	Panjang	Keterangan	Key
ID_Rule	Text	4	No rule / aturan	PK
P1	Text	-	Identifikasi awal	
P2	Text	-	Kerusakan mesin	
P3	Text	-	Kesalahan metode	
P4	Text	-	Kesalahan operator	
P5	Text	-	Kondisi material	
S1	Text	-	Nama defect	
S2	Text	-	Proses kritis	
Solusi	Memo	-	Deskripsi solusi	
Fire	Number	-	Penandaan / bookmark	

5.4.2 Perancangan *Decision Table* (Tabel Keputusan)

Pembuatan *decision table* dilakukan berdasarkan *dependency diagram* yang telah dibuat sebelumnya. *Decision table* diperlukan untuk menunjukkan hubungan antara nilai-nilai dengan hasil atau keluaran. Berikut ini adalah contoh *rule*/aturan penelusuran cacat produk pada pianika berdasarkan macam identifikasi awal, kerusakan mesin, kesalahan metode, kesalahan operator, dan kondisi material yang dapat dilihat pada Tabel 5.16.

Dari berbagai kemungkinan jawaban yang mungkin diberikan, tidak semua bisa menghasilkan rekomendasi akhir karena ada beberapa yang tidak masuk akal. Dalam hal tersebut, maka perlu dilakukan reduksi untuk memilih *rule* yang bisa dipakai dan menghasilkan rekomendasi pakar.

Tabel 5.16. Tabel *Rule Decision*

Rule/aturan
<p>Rule Set 1</p> <p>ID_Rule: R001</p> <p>IF P1 = Tiup Dua Nada Sama, grafik osiloscope tidak berhenti</p> <p>AND IF P2 = Mesin tuning tidak stabil / rusak</p> <p>AND IF P3 = Kikiran pangkal melebar High / Low</p> <p>AND IF P4 = Bekerja tidak sesuai IK</p> <p>AND IF P5 = Penampilan reed keseluruhan turun</p> <p>THEN S1 = Oktaf aneh</p> <p>S2 = Proses Reed Plate, Proses Tuning</p> <p>Solusi = 1. Cek riwayat dan umur mesin, periksa apa bisa diperbaiki atau ganti dengan mesin yang baru. Apabila umur mesin lebih dari 5 tahun, sebaiknya ganti dengan yang baru.</p> <p>2. Pada saat pengikiran pangkal diusahakan seluruh permukaan reed dikikir dengan penekanan yang sama/merata</p> <p>3. Sosialisasi Instruksi Kerja kepada operator agari dan tuning</p> <p>4. Lakukan pengecekan pada proses tuning sebelum pianika dituning.</p>

5.4.3 Desain *Inference Engine*

Setelah struktur knowledge base yang berupa tabel-tabel terpetakan, maka algoritma *inference engine* dapat di rancang. Berikut ini adalah bentuk pseudo code algoritmanya:

Tabel 5.17. *Pseudo Code Inference Engine*

```
Mulai program
Buka tabel Input
Set pointer pada baris pertama tabel Input
Buat loop
    Ambil fakta-fakta faktor penyebab pada baris pertama
    Set pointer pada baris pertama tabel Decision
    Buka tabel Decision
        Ambil semua premise pada baris pertama
        Jika fakta cocok dengan premise, maka
            Tandai kolom Fire (Meledak) dengan angka 1
    Loop baris selanjutnya pada tabel Decision
    Letakkan hasil pada tabel sementara
Loop baris selanjutnya pada tabel Input
Tutup tabel Input
Tutup tabel Decision
```

5.5 Implementasi

Tahap implementasi sistem adalah tahap untuk menerapkan semua desain sistem yang telah dibuat ke dalam bentuk program aplikasi berbasis komputer. *Software* yang digunakan untuk implementasi dalam hal ini adalah *Microsoft Access 2007* dan *qc-toolbox*.

5.5.1 *Input Data Audit*

Input data audit merupakan proses utama dalam aplikasi ini. Proses dimulai dengan memasukkan nama, tanggal audit, jumlah produksi aktual harian, dan jumlah cacat produk sesuai dengan kategori melalui *form* audit data tiap produk. Contoh *form*

audit data untuk produk pianika P32 dapat dilihat pada Gambar 5.7. Dari *form* tersebut, dengan hanya memasukkan input data yang telah disebutkan sebelumnya, dapat dilihat juga Persentasi barang yang tidak baik, jumlah barang yang tidak baik (defect), dan diagram batang harian.



Gambar 5.7 Form Audit Data Pianika P32

5.5.2 Statistical Quality Control

Alat pengendalian kualitas yang dipakai pada aplikasi ini adalah Peta Kontrol np, *Pareto Diagram* dan *Fishbone Diagram*. Setelah *user* memasukkan data audit produk, *user* dapat langsung melihat *Pareto Diagram* dan Peta Kontrol secara otomatis. Sedangkan *Fishbone Diagram* digunakan untuk hasil *expert system*. Untuk dapat mengakses SQC tersebut, *user* harus memasukkan data bulan dan tahun sebagai parameter pembuatan SQC pada *form* SQC yang digambarkan pada Gambar 5.9.

BuatChart

STATISTICAL QUALITY CONTROL

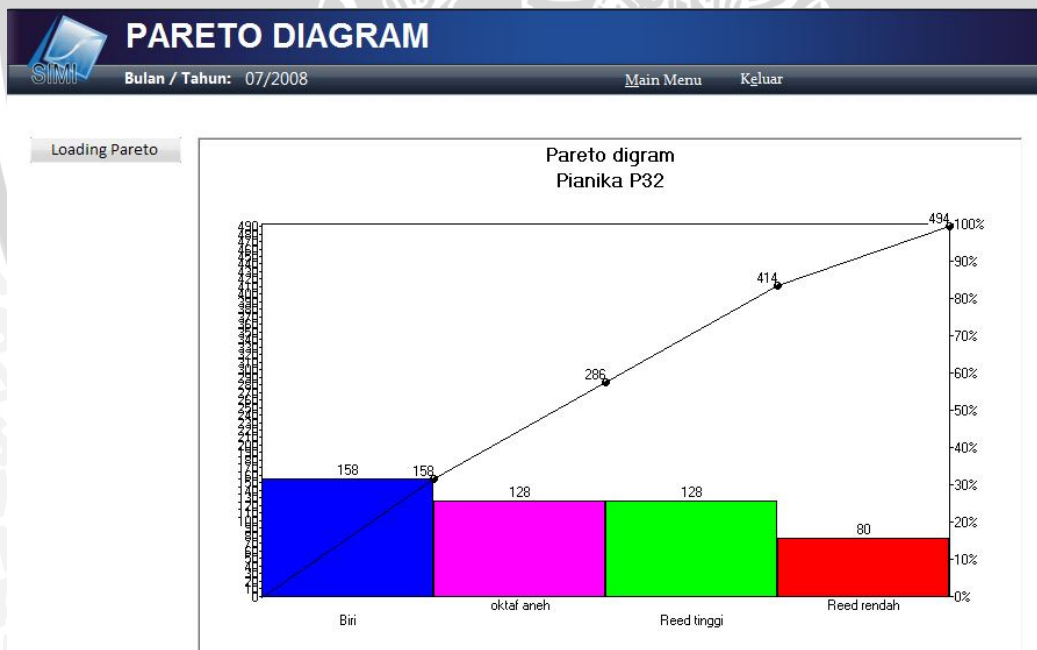
PRODUK PIANIKA Main Menu Keluar

Pianika P25
Bulan/Tahun: Peta Kontrol
Pareto

Pianika P32
Bulan/Tahun: Peta Kontrol
Pareto

Pianika P37
Bulan/Tahun: Peta Kontrol
Pareto

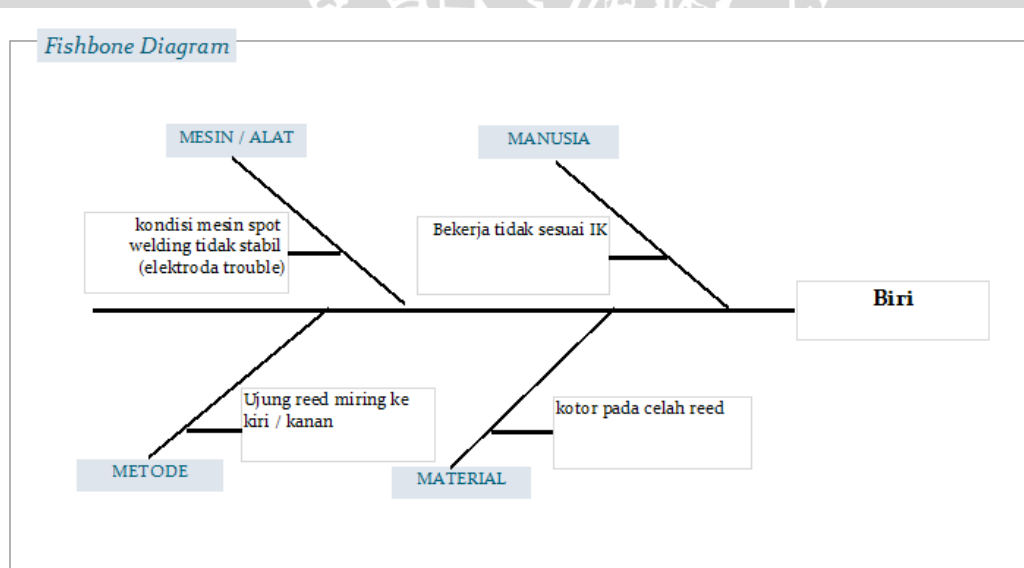
Gambar 5.8 Form SQC



Gambar 5.9 Form Pareto Diagram Pianika P32



Gambar 5.10 Form Peta Kontrol Pianika P32



Gambar 5.11 Fishbone Diagram untuk cacat produk Biri

Dari Gambar 5.9 tersebut dapat dilihat Pareto Diagram untuk cacat produk pianika P32 sekaligus nilai kumulatifnya dalam satu bulan. Pada Gambar 5.10 dapat dilihat peta kontrol pianika P32 beserta data-datanya untuk selanjutnya dapat dilakukan

analisa pengendalian kualitas. Pada Gambar 5.11 merupakan gambar hasil proses *expert system*, yaitu berupa *fishbone* diagram sebagai salah satu alat pengendalian kualitas.

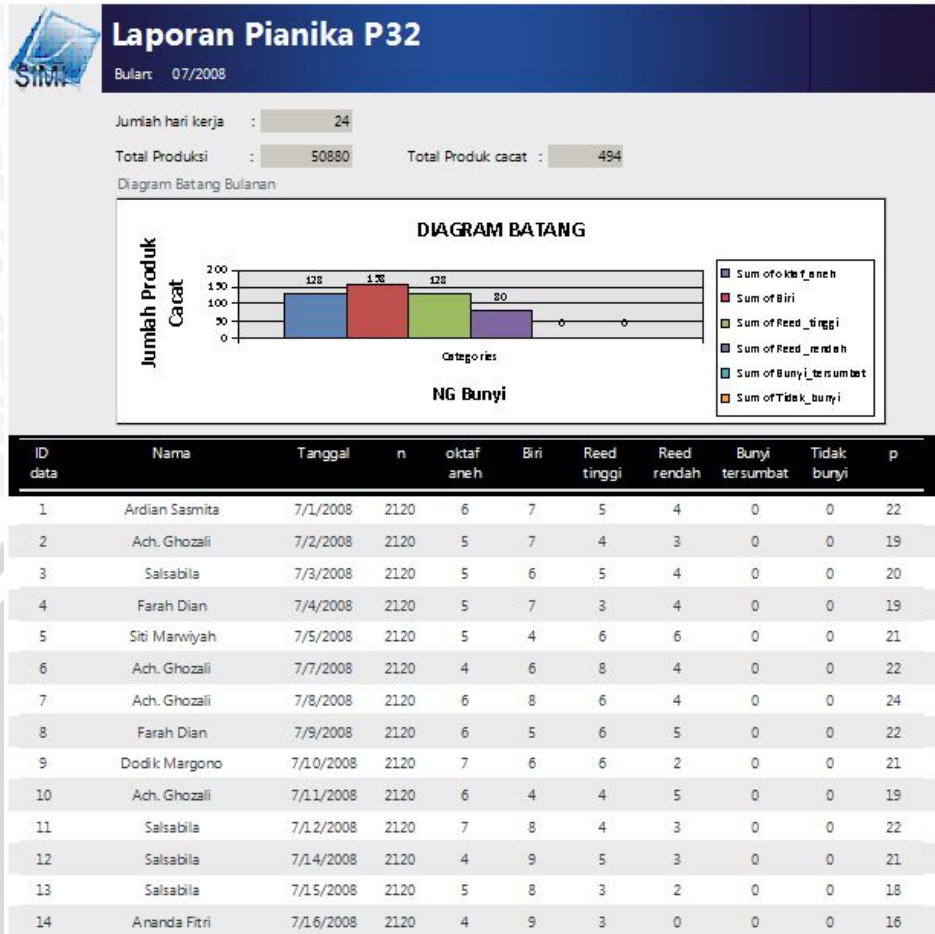
5.5.3 Laporan / Report

Pembuatan laporan dalam aplikasi ini dibagi menjadi 2 periode: bulanan dan tahunan. Untuk laporan data pegawai merupakan laporan berbentuk *summary*. Untuk dapat membuat laporan, *user* harus memasukkan data bulan dan tahun atau tahun saja pada *form* Buat Laporan yang digambarkan pada Gambar 5.12.

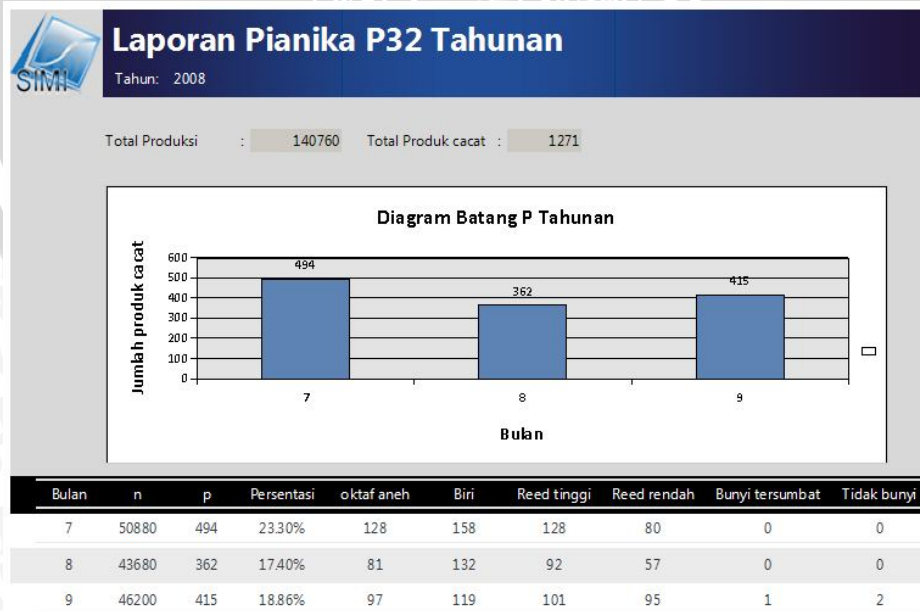
The screenshot shows a web-based application window titled "LAPORAN PIANIKA". At the top right, there are two buttons: "Main Menu" and "Keluar". The main content area is divided into three sections, each for a different report type:

- Laporan Pianika P25:** Includes a "Bulanan (mm/yyyy) :" input field with a "Print" button, and a "Tahunan (yyyy) :" input field with a "Print" button.
- Laporan Pianika P32:** Includes a "Bulanan (mm/yyyy) :" input field with a "Print" button, and a "Tahunan (yyyy) :" input field with a "Print" button.
- Laporan Pianika P37:** Includes a "Bulanan (mm/yyyy) :" input field with a "Print" button, and a "Tahunan (yyyy) :" input field with a "Print" button.

Gambar 5.12 *Form* Laporan Pianika



Gambar 5.13 Form Laporan Pianika Bulanan



Gambar 5.14 Form Laporan Pianika Tahunan


5.5.4 Expert System

Expert system dalam aplikasi ini juga merupakan proses utama selain proses *input* audit data produk. Sebelum *user* memulai program *expert system*, *user* dihadapkan sebuah jendela *pop up*. Jendela *pop up* disini, berisi informasi penting sebelum menjalankan program *expert system*. Setelah melanjutkan untuk menjalankan *expert system*, *user* harus memasukkan beberapa data informasi antara lain: tanggal, nama, identifikasi awal / gejala, kerusakan mesin, kesalahan metode, kesalahan operator dan kondisi material. Apabila semua data telah terisi, *user* dapat menekan tombol Solusi dan melihat solusi yang dihasilkan *expert system*. Apabila salah satu data belum diisi, maka tombol Solusi tidak akan muncul dan program tidak dapat dijalankan. Beberapa *form* yang berkaitan dengan proses *expert system* dapat dilihat sebagai berikut:



EXPERT SYSTEM			
ID_input	23	Tanggal	1/1/2010
Nama	Farah Quinn		
Identifikasi Awal	Tiup nada tunggal, grafik osiloscope tidak berhenti		
Ciri Kerusakan Mesin	kondisi mesin spot welding tidak stabil (elektroda trouble)		
Metode	Ujung reed miring ke kiri / kanan		
Operator	Bekerja tidak sesuai IK		
Ciri Material	kotor pada celah reed		
Solusi			

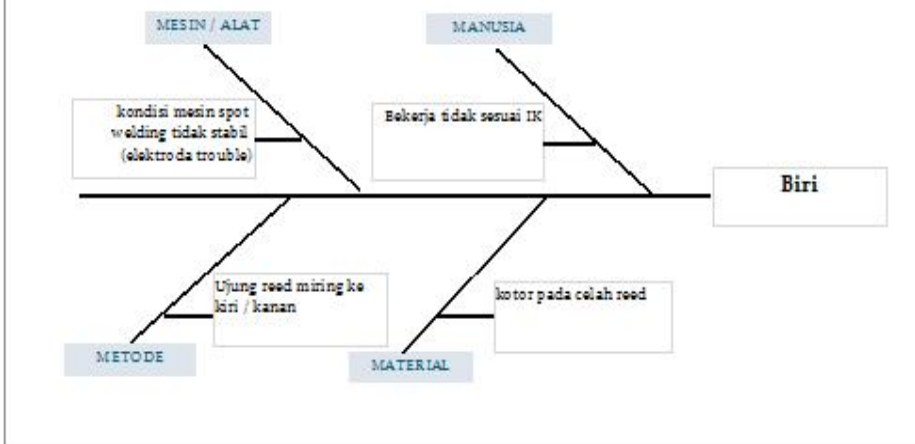
Gambar 5.15 Form Input Expert System


EXPERT SYSTEM
 Give you solutions

Nama	Farah Quinn
Tanggal	1/1/2010

Identifikasi Awal Tiup nada tunggal, grafik osiloscope tidak berhenti

Fishbone Diagram



Proses Kritis	Proses Reed plate, agari, tuning
Solusi	<ol style="list-style-type: none"> 1. Ganti elektroda yang baru pada mesin spot welding 2. Reed ditarik lurus dari pangkal dan dipuntir berlawanan arah, digeser sedikit sampai center, dilakukan penekanan untuk melihat arah gerak reed dengan menggunakan spanner 3. Sosialisasi Instruksi Kerja kepada operator agari dan tuning 4. kotor pada celah reed di sebabkan oleh sisa gram yang belum disikat pada proses tuning, sikatlah setiap reed setelah melakukan pengikiran

Gambar 5.16 Hasil *Expert System*

5.6 Pengujian Software

Pengujian program aplikasi SIPKES ini menggunakan dua segi, yaitu verifikasi dan validasi. Verifikasi bertujuan untuk menguji apakah *software* aplikasi sudah sesuai dengan desain sistem yang telah dibuat sebelumnya. Hal ini berguna untuk menyeleraskan *software* hasil implementasi agar tidak menyimpang dari desain sistemnya. Sedangkan validasi bertujuan untuk menguji apakah sistem informasi pengendalian kualitas dan expert system yang telah dibuat dapat berfungsi sepenuhnya sebagai sistem informasi yang mampu memberikan informasi yang diharapkan.

5.6.1 Personel yang Melakukan Pengujian

Pengujian program dilakukan oleh :

1. Ir. Purnomo Budi S., M.Sc., PhD dan Nasir W. Setyanto., ST., MT selaku dosen pembimbing tugas akhir.
2. Anggun Puspita, penulis atau perancang prototipe.
3. Iwan Moestanto, Manager Departemen PCR di PT. Yamaha Musical Products Indonesia, selaku pakar.

5.6.2 Uji Verifikasi

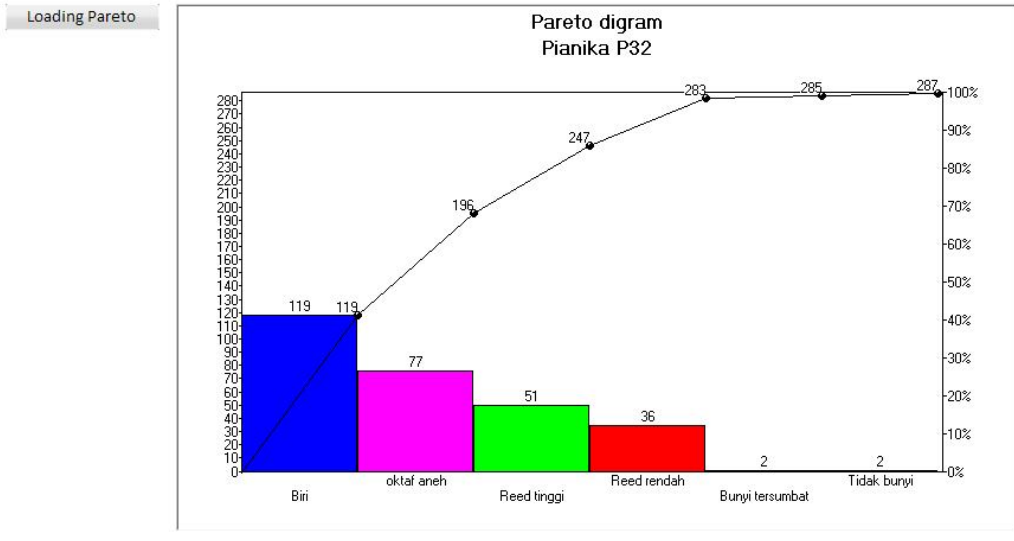
Uji verifikasi dilakukan untuk mengetahui apakah apakah program berfungsi dengan baik dan sesuai dengan desain yang direncanakan. Hal itu dilakukan dengan memeriksa apakah ada kesalahan dalam analisa, *control program*, *input* dan *output program*. Program dijalankan melalui aplikasi *Microsoft Access 2003*. Salah satu contoh verifikasi yang terdapat dalam aplikasi ini adalah verifikasi *user interface* untuk fungsi matematis peta kontrol dan *pareto diagram*. Secara acak yang dipilih adalah produk pianika P25. Adapun langkah-langkah uji verifikasi adalah sebagai berikut :

- a. Jalankan program
- b. Buka form SQC
- c. Masukkan bulan dan tahun kemudian klik tombol Peta Kontrol



Gambar 5.17 Hasil Uji Verifikasi Peta Kontrol

- d. Kembali ke form SQC
- e. Masukkan bulan dan tahun kemudian klik tombol Pareto



Gambar 5.18 Hasil Uji Verifikasi *Pareto Diagram*

Verifikasi untuk fungsi matematis perhitungan atribut peta kontrol dan pareto diagram dilakukan dengan cara membandingkan hasil perhitungan oleh aplikasi dengan hasil perhitungan manual menggunakan kalkulator. Berikut perhitungan manual:

a. Perhitungan manual atribut Peta Kontrol

$\bar{p} = p / (24 \times n)$	$CL = n \bar{p}$
$= 287 / (24 \times 170)$	$= 170 \times 0.0703$
$= 0.0703$	$= 11.95$

$$UCL = \bar{p} + 3\sqrt{n\bar{p}(1-\bar{p})}$$

$$= 11.95 + 3\sqrt{11.95(1-0.0703)}$$

$$= 21.951$$

$$LCL = \bar{p} - 3\sqrt{n\bar{p}(1-\bar{p})}$$

$$= 11.95 - 3\sqrt{11.95(1-0.0703)}$$

$$= 1.946$$

b. Perhitungan manual atribut *Pareto Diagram*Tabel 5.18 Perhitungan Manual Atribut *Pareto Diagram*

Jenis Cacat	Total	Kumulatif Total	% Cacat	% Kumulatif
Biri	119	119	41.63	41.63
Oktaf aneh	77	196	26.82	68.45
Reed tinggi	51	247	17.8	86.22
Reed rendah	36	283	12.54	98.76
Bunyi tersumbat	2	285	0.07	99.46
Tidak bunyi	2	287	0.07	100.00

Berdasarkan perhitungan manual, dapat dilihat bahwa fungsi matematis untuk perhitungan atribut peta kontrol dan *pareto diagram* aplikasi SIPKES adalah benar. Hal ini dibuktikan dengan samanya hasil perhitungan aplikasi dengan hasil perhitungan secara manual.

5.6.3 Uji Validasi

Uji validasi dilakukan dengan tujuan untuk mengetahui apakah solusi yang dihasilkan *Expert System* sama dengan yang direncanakan. Berikut salah satu contoh pengujian:

Diberikan data:

1. Jika ditiup dengan tekanan rendah atau ditiup dengan lemah, suaranya terlambat keluar
2. Chucking trouble pada mesin spot welding
3. Terlalu mengangkat waktu memasukkan spatula ketika mengikir
4. Bekerja tidak sesuai IK
5. kotor pada celah reed

Input data pada program aplikasi adalah ditunjukkan pada Gambar 5.20 dan hasil program ditunjukkan pada Gambar 5.21. Hasil expert system memberikan rekomendasi solusi dan proses kritis yang perlu perbaikan. Solusi tersebut adalah

1. Pastikan baut pada chucking yang terpasang dikanan dan dikiri sudah longgar, Pasang JIG diantara Chucking dan Kiridasi kemudian masukkan jig ke mulut chucking, Kencangkan baut yang ada di sebelah kanan dan kiri chucking dan jangan lupa memasang back up chucking
2. Turunkan reed setelah proses tuning (choiritsu) sesuai standar
3. Sosialisasi Instruksi Kerja kepada operator agari dan tuning
4. Digeser dengan menggunakan spanner sehingga reed jadi center

Rule yang bekerja adalah:

Tabel 5.19. *Rule* / Aturan yang Bekerja Pada Uji Validasi

Rule/aturan
Rule Set 18
ID_Rule: R018
IF P1 = Jika ditiup dengan tekanan rendah atau ditiup dengan lemah, suaranya terlambat keluar
AND IF P2 = Chucking trouble pada mesin spot welding
AND IF P3 = Terlalu mengangkat waktu memasukkan spatula ketika mengikir
AND IF P4 = Bekerja tidak sesuai IK
AND IF P5 = kotor pada celah reed
THEN S1 = Reed Tinggi
S2 = Proses Reed Plate, Proses Tuning
Solusi = 1. Pastikan baut pada chucking yang terpasang dikanan dan dikiri sudah longgar, Pasang JIG diantara Chucking dan Kiridasi kemudian masukkan jig ke mulut chucking, Kencangkan baut yang ada di sebelah kanan dan kiri chucking dan jangan lupa memasang back up chucking
2. Turunkan reed setelah proses tuning (choiritsu) sesuai standar
3. Sosialisasi Instruksi Kerja kepada operator agari dan tuning
4. Digeser dengan menggunakan spanner sehingga reed jadi center

Dari hasil uji validasi memperlihatkan bahwa hasil solusi yang direkomendasikan program sesuai dengan pakar dan perencanaan. *Expert System* juga dapat berjalan dengan baik.

EXPERT SYSTEM
Baru Iutup

ID_input	25	Tanggal	2/2/2010
Nama	Iwan Moestanto		
Identifikasi Awal	Jika ditiup dengan tekanan rendah atau ditiup dengan lemah, suaranya terlambat keluar		
Ciri Kerusakan Mesin	Chucking trouble pada mesin spot welding		
Metode	Terlalu mengangkat waktu memasukkan spatula ketika mengikir		
Operator	Bekerja tidak sesuai IK		
Ciri Material	kotor pada celah reed		

Solusi

Gambar 5.19 Input Data Uji Validasi *Expert System*

EXPERT SYSTEM
Give you solutions

Nama	Iwan Moestanto
Tanggal	2/2/2010

Identifikasi Awal Jika ditiup dengan tekanan rendah atau ditiup dengan lemah, suaranya terlambat keluar

Fishbone Diagram

Proses Kritis	Proses Reed plate, Tuning
Solusi	<ol style="list-style-type: none"> Pastikan baut pada chucking yang terpasang dikanan dan dikiri sudah longgar, Pasang JIG diantara Chucking dan Kiridas kemudian masukkan jig ke mulut chucking. Kencangkan baut yang ada di sebelah kanan dan kiri chucking dan jangan lupa memasang back up chucking Turunkan reed setelah proses tuning (choiritsu) sesuai standar Sosialisasi Instruksi Kerja kepada operator agari dan tuning Digeser dengan menggunakan spanner sehingga reed jadi center

Gambar 5.20 Hasil Uji Validasi *Expert System*

BAB VI

PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat diperoleh setelah melakukan proses Perancangan Sistem Informasi Pengendalian Kualitas dan *Expert System* antara lain:

1. Integrasi Sistem Informasi Pengendalian Kualitas dan *Expert System* ke dalam sebuah prototipe aplikasi sangat membantu divisi produksi dalam perusahaan untuk mendapatkan solusi tentang cacat produk secara cepat dan tepat. Selain itu, sistem juga dapat memberi informasi mengenai proses kritis yang perlu perbaikan dan hubungan sebab akibat cacat produk melalui visualisasi *fishbone diagram*.
2. Program aplikasi protoipe yang telah dirancang dapat menggantikan sistem pengendalian kualitas produk secara manual menjadi sistem pengendalian kualitas berbasis komputer. Program aplikasi yang telah didukung dengan peta kontrol dan *pareto diagram* serta diagram batang yang dapat digunakan setiap saat akan memudahkan operator untuk dapat menganalisa pengendalian kualitas secara cepat dan efisien. Hal ini tentu saja dapat mempercepat pengontrolan produk-produk *defects*.

6.2 Saran

Adapun Perancangan Sistem Informasi Pengendalian Kualitas dan *Expert System* yang dihasilkannya masih memiliki beberapa kekurangan. Oleh karena itu ada beberapa saran yang dianjurkan, antara lain:

1. SIPKES dapat diintegrasikan dengan *line* produksi, sehingga operator dengan mudah mengetahui *line* mana yang kritis dan perlu perbaikan proses.
2. SIPKES ini masih dalam tahap prototipe, oleh karena itu masih diperlukan pengembangan dan penelitian lebih lanjut apabila ingin diaplikasikan dalam suatu perusahaan.

DAFTAR PUSTAKA

- Anonymous. 2003. *Pengembangan Sistem Pakar Menggunakan Visual Basic*. Yogyakarta: Andi.
- C. Kenneth, Laudon & P. Jane, Laudon. 2005. *Sistem Informasi Manajemen, Mengelola Perusahaan Digital*. Yogyakarta: Andi.
- Dologite, D.G. 1993. *Developing Knowledge-Based Systems Using VP-Expert*. New York: Macmillan Publishing Company.
- Durkin, Jhon. 1996. *Expert System Design and Development*. USA: Prentice Hall International Inc.
- F. Luger, George & A. Stubblefield, William. 2000. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* 3rd edition. Addison-Wesley.
- Fathansyah, Ir. 1999. *Basis Dat*. Bandung: Informatika.
- Gaspersz, Vincent, Dr., M.St, CIQA, CPIM. 2003. *Statistical Process Control: Penerapan Teknik-Teknik Statistikal dalam Manajemen Bisnis Total*. Yayasan Indonesia Emas, Institut VINCENT. Jakarta : PT Gramedia Pustaka Utama,.
- Giarratano & Riley. 1989. *Expert System Principles and Programming*. Boston: PWSKENT Publishing Company.
- Grant, E.L. &Leavenworth, R.S. 1999. *Statistical Quality Control*, 7th ed. New York: McGraw-Hill.
- Gyrna, Frank M. 2001. *Quality Planning and Analysis: From Product Development through Use*. 4th ed. New York : McGraw-Hill.
- Harmon, Paul & Sawyer, Brian. 1990. *Creating Expert Systems for Business and Industry*. New York: John Wiley & Sons, Inc.
- Kolarik, W.J. 1999. *Creating Quality: Process Design For Results*, New York: McGraw-Hill.
- Kroenke, D.M. 2005. *Database Processing: Dasar-Dasar, Desain dan Implementasi*. Jakarta : Erlangga.
- Montgomery, D.C. 2001. *Introduction to Statistical Quality Control*. 4th ed., New York: John Wiley and Sons Inc.
- Sutedjo, Budi & Oetomo, Dharma, S.Kom., MM. 2002. *Perancangan dan Pembangunan Sistem Informasi*. Yogyakarta: Andi.

Lampiran 1. Manual Prototipe

MANUAL PROTOTIPE

1. Pendahuluan

Aplikasi Sistem Informasi Pengendalian Kualitas dan *Expert System* adalah aplikasi sistem informasi yang berguna untuk mencatat, mengorganisasikan, dan mengolah data-data produksi, audit produk dan cacat produk, sehingga dapat memberikan informasi yang akurat kepada user dan dapat membantu dalam pengendalian kualitas serta mendapatkan solusi atas kecacatan produk.

2. Kebutuhan Sistem

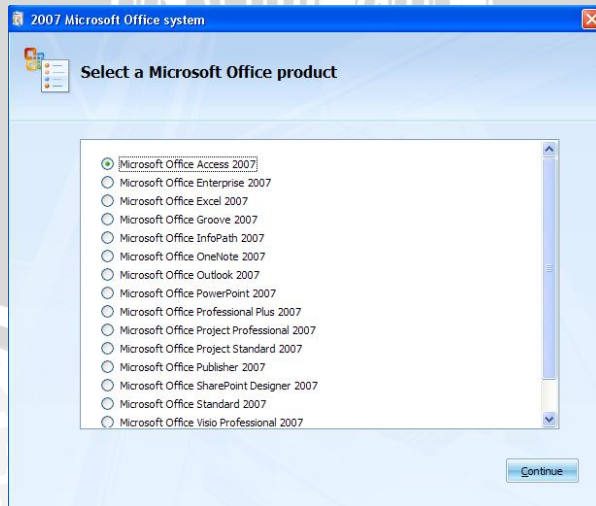
Untuk dapat menjalankan aplikasi SIPKES, ada beberapa kebutuhan sistem yang harus dipenuhi berkaitan dengan spesifikasi *hardware* dan *software*, antara lain:

- Prosesor minimal *Pentium IV 450 MHz*, atau yang lebih tinggi.
- *RAM 512 MB* atau lebih.
- Sistem operasi *Windows Windows XP, Windows Vista, Windows Seven*.
- Kapasitas *harddisk* yang tersisa minimal *100 MB*.

3. Petunjuk Instalasi

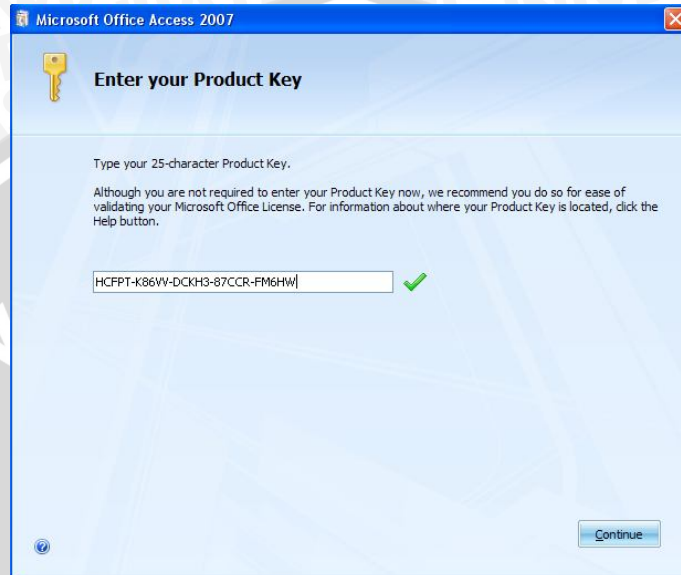
Untuk dapat menjalankan aplikasi SIPKES ini, dalam komputer klien harus sudah terinstall program *Microsoft Access 2007* terlebih dahulu. Cara melakukan instalasi *Micosoft Access 2007* yaitu:

- Masukkan *CD Microsoft Office 2007* ke dalam *CDROM*. Ikuti petunjuk pada kotak dialog, hingga muncul dialog seperti gambar 1.1.



Gambar 1.1 Langkah Awal Instalasi *Microsoft Office 2003*.

- User dapat memilih mana aplikasi yang akan diinstall dengan memilih option pilihan yang tersedia. Pilihlah aplikasi *Microsoft Office Access 2007*. Setelah selesai, klik *Continue*.
- Masukkan 25 karakter *serial number* yang biasa tersedia di cover CD atau dalam bentuk *notepad*. Setelah selesai, klik *Continue*.



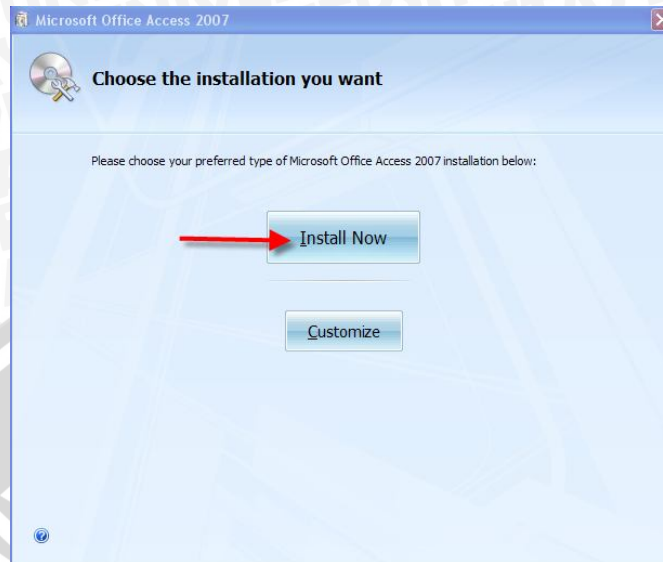
Gambar 1.2 *Serial Number Installations.*

- Selanjutnya akan ditampilkan kotak dialog *Microsoft Software License Terms*. Beri tanda centang pada kotak “*I accept the terms of the agreement*”. Setelah selesai, klik *Continue*.



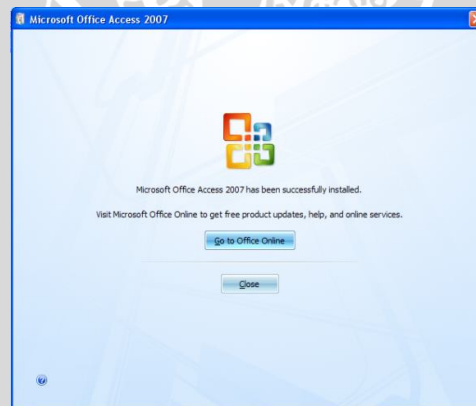
Gambar 1.3 *Microsoft Software License Terms*

- Selanjutnya Pilih *Install Now* untuk memulai penginstalan *Microsoft Access 2007*.



Gambar 1.4 Memulai Instalasi.

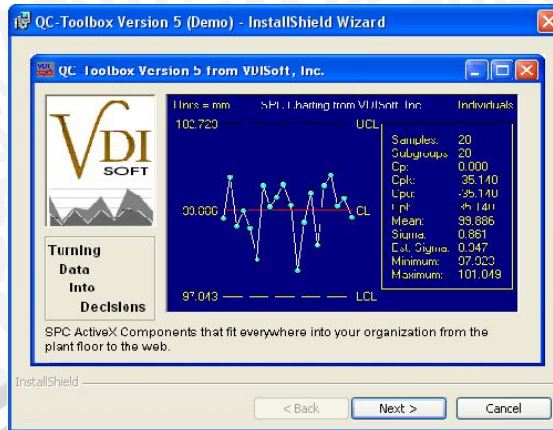
- Tunggu hingga proses instalasi selesai. Apabila instalasi berhasil dilakukan, maka akan tampil kotak dialog seperti gambar 1.5. Klik *Close* untuk menutup kotak dialog.



Gambar 1.5 Dialog *Install Complete*.

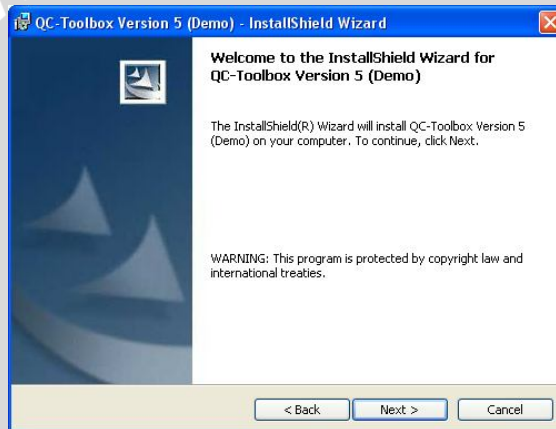
Langkah selanjutnya adalah instalasi *Add-ins ActiveX Control* yang fungsinya adalah untuk membaca grafik peta kontrol dan *pareto* dalam aplikasi SIPKES. Berikut Cara instalasinya:

- Pilih *file qctbx5d.exe* pada direktori *file* yang bersamaan dengan *file* aplikasi SIPKES. Klik dua kali pada *file* aplikasi tersebut. Maka akan muncul dialog *InstallShield Wizards* seperti gambar di bawah ini, selanjutnya klik *Next*:



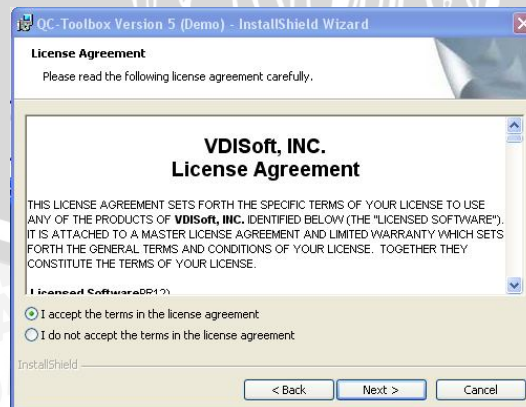
Gambar 1.6 Dialog *InstallShield Wizards*.

- Setelah itu akan muncul kotak dialog berupa informasi dan peringatan seperti gambar dibawah ini, klik *Next* untuk melanjutkan instalasi:



Gambar 1.6 Dialog *QC-Toolbox Version 5*.

- Berikutnya akan muncul kotak dialog *License Agreement*. Pilih *I accept the terms in the license agreement*. Setelah selesai, klik *Next*.



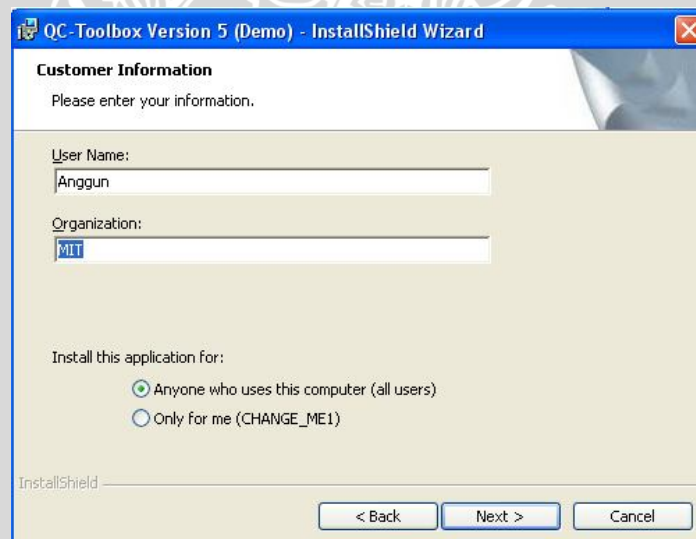
Gambar 1.7 *License Agreement QC-Toolbox*.

- Selanjutnya akan muncul kotak dialog informasi yang berkaitan dengan aplikasi lain. Klik *Next* untuk melanjutkan.



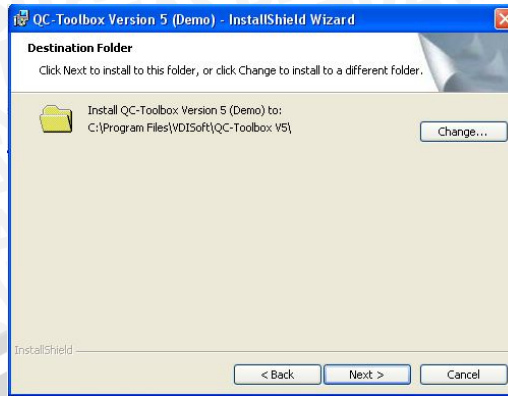
Gambar 1.8 *Readme Information QC-Toolbox.*

- Masukkan *username* dan *Organization*. Setelah selesai klik *Next*.



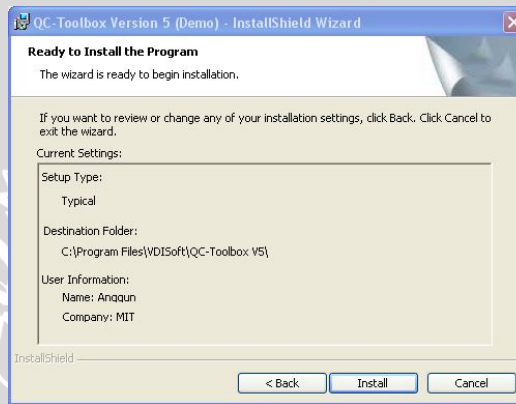
Gambar 1.9 *Customer Information*

- Pilih *destination folder* untuk yang akan digunakan, secara *default* computer akan meletakkan pada *folder Program Files* di *drive C*. Setelah selesai klik *Next*.



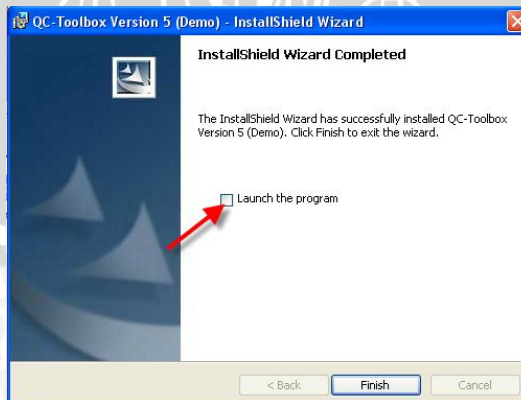
Gambar 1.10 Destination Folder QC-Toolbox

- Berikutnya akan muncul kotak dialog *summary* instalasi. Klik *install* untuk memulai instalasi.



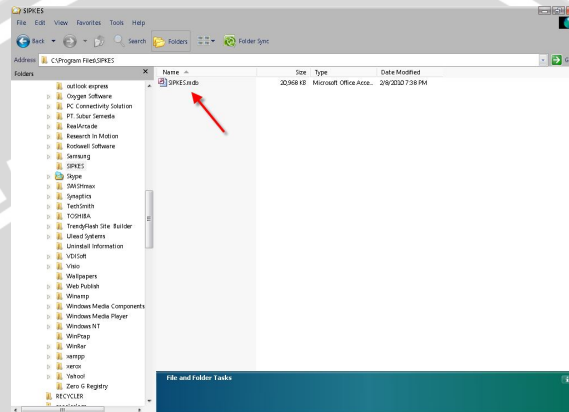
Gambar 1.11 Summary QC-Toolbox Installation

- Tunggu proses instalasi selesai hingga muncul kotak dialog seperti gambar di bawah ini. Hilangkan centang untuk “*Launch The Program*”, kemudian klik *Finish*.



Gambar 1.12 QC-Toolbox Installation Completed

- Langkah selanjutnya adalah mengcopy file SIPKES agar dapat dijalankan. Buat folder baru di dalam *Program Files* dengan melakukan klik kanan, pilih *new*, pilih *folder*.
- Setelah itu ganti namanya dengan SIPKES, klik dua kali untuk masuk ke dalam *folder*.
- Copykan file SIPKES dengan mempastenya ke dalam *folder*, kemudian klik dua kali file SISDD untuk menjalankannya.

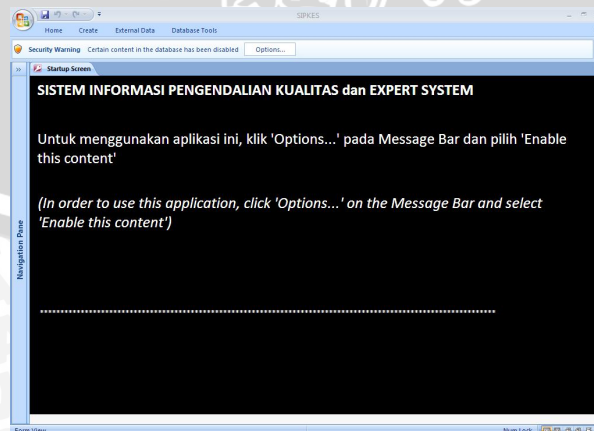


Gambar 1.13 Menjalankan SIPKES

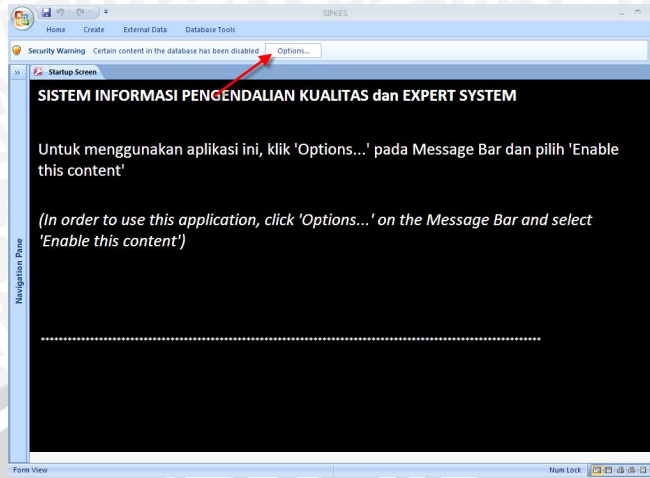
4. Petunjuk Pengoperasian

Program aplikasi SISDD dapat dijalankan dengan mengklik dua kali file SIPKES yang telah dicopy. Setelah itu langkah-langkahnya adalah sebagai berikut.

- Tunggu hingga muncul *startup screen* seperti gambar 1.14. *startup screen* berisi petunjuk awal sebelum menggunakan aplikasi SIPKES. Lakukan sesuai petunjuk pada *startup screen*, yaitu klik *options* pada *message bar* dan pilih '*Enable this content*'.



Gambar 1.14 Startup Screen SIPKES.

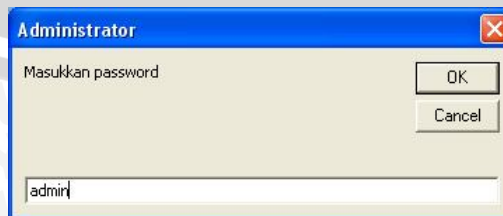


Gambar 1.15 Pilih *Options*.



Gambar 1.16 Pilih *Enable this content*.

- Setelah itu akan muncul kotak dialog untuk *administrator*. Masukkan *password* yaitu: “admin” (tanpa tanda kutip) apabila ingin mengakses aplikasi sebagai admin atau dapat juga langsung klik *ok* atau *cancel* apabila ingin masuk sebagai *Guest*. Apabila masuk sebagai *Guest* ada sebuah menu yang tidak dapat diakses, hanya admin yang dapat mengakses.



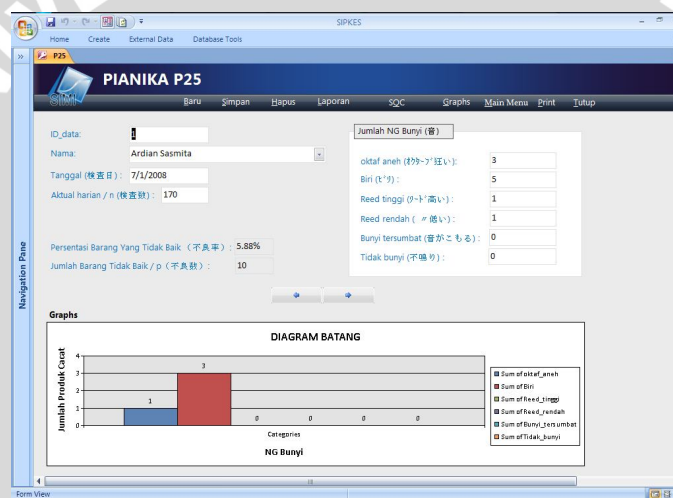
Gambar 1.17 Kotak Dialog Administrator

- Setelah klik *ok*, maka akan muncul main menu aplikasi.



Gambar 1.18 Main Menu SIPKES

- Pilih menu program yang diinginkan, apabila ingin mengaudit data produk. Klik pada menu Edit Data, kemudian pilih produk yang akan di audit.



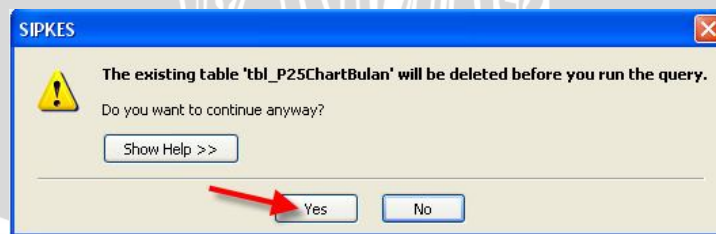
Gambar 1.19 Audit Data Pianika P25

- Form ini mempunyai beberapa tombol antara lain:
 - Baru, berfungsi untuk menambah data baru
 - Simpan, berfungsi untuk menyimpan data yang telah dimasukkan
 - Hapus, berfungsi untuk menghapus data
 - Laporan, berfungsi untuk membuka form membuat laporan
 - SQC, berfungsi untuk membuka form SQC
 - Graphs, berfungsi untuk membuka form diagram batang cacat produk
 - Main Menu, berfungsi untuk membuka form main menu
 - Print, berfungsi untuk mencetak form
 - Tutup, berfungsi untuk menutup form
 - Tombol panah ke kiri, berfungsi untuk melihat data sebelumnya

- Tombol panah ke kanan, berfungsi untuk melihat data berikutnya
- Untuk membuat Peta kontrol serta *Pareto Diagram*, bukalah *form SQC*. Dapat diakses melalui *main menu* ataupun dari audit data produk. Masukkan bulan dan tahun sesuai format yang telah disediakan pada *text box* sesuai dengan jenis produk yang diinginkan, kemudian pilihlah *SQC* yang diinginkan. Klik tombol bertuliskan peta kontrol untuk membuat peta kontrol, atau tombol yang bertuliskan *Pareto* untuk membuat *Pareto diagram*.

Gambar 1.20 *Form Statistical Quality Control*

- Setelah itu akan muncul kotak dialog konfirmasi untuk menghapus data pada suatu tabel. Klik tombol 'yes' untuk melanjutkan. Apabila memilih 'no', maka proses pembuatan peta kontrol tidak dapat dijalankan.



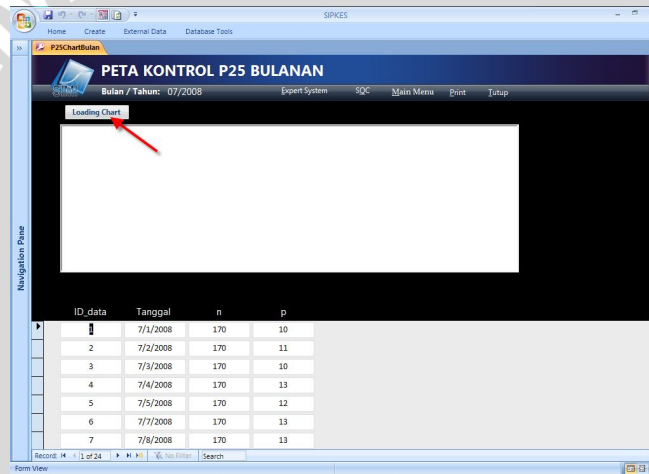
Gambar 1.21 Kotak Dialog Konfirmasi

- Kemudian akan muncul juga *pop up window QC-Toolbox*, Klik *Close* untuk menutup dan melanjutkan pembuatan peta kontrol.



Gambar 1.22 Pop Up Window QC-Toolbox

- Pada Form Peta Kontrol, klik tombol *loading chart* untuk menampilkan peta kontrol. Setelah itu akan muncul peta kontrol pada bulan yang diinginkan.



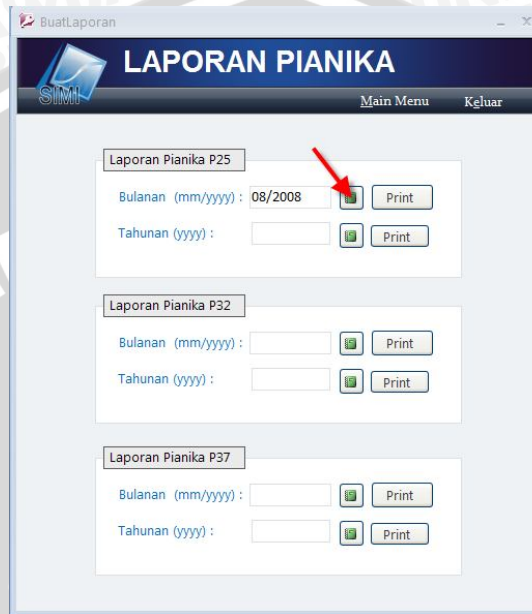
Gambar 1.23 Klik Tombol *Loading Chart*

- Untuk membuat laporan, akses dapat melalui main menu. Klik tombol Laporan seperti gambar di bawah ini:



Gambar 1.24 Klik Tombol Laporan

- Kemudian akan muncul kotak *form* Laporan Pianika. Masukkan Bulan dan Tahun untuk pembuatan laporan bulanan atau tahun saja untuk membuat laporan tahunan pada *text box* masing-masing jenis produk. Setelah itu klik tombol buku berwarna hijau untuk melihat laporan dan untuk mencetaknya, klik tombol *Print*.



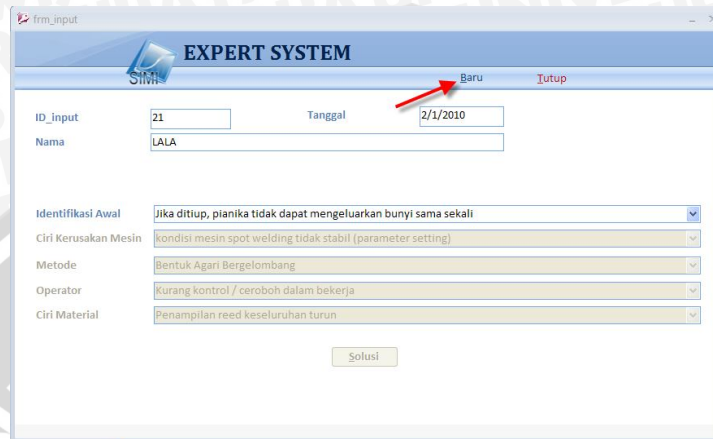
Gambar 1.25 Klik Tombol Melihat Laporan

- Untuk dapat menjalankan *expert system*, user dapat mengakses melalui main menu. Klik tombol *Expert System*.



Gambar 1.26 Klik Tombol *Expert System*

- Selanjutnya akan muncul pop up window expert system yang berisi petunjuk sebelum menjalankan expert system. Klik 'Lanjutkan' untuk melanjutkan. Setelah itu, akan muncul form expert system. Untuk memulai, klik tombol 'Baru'.



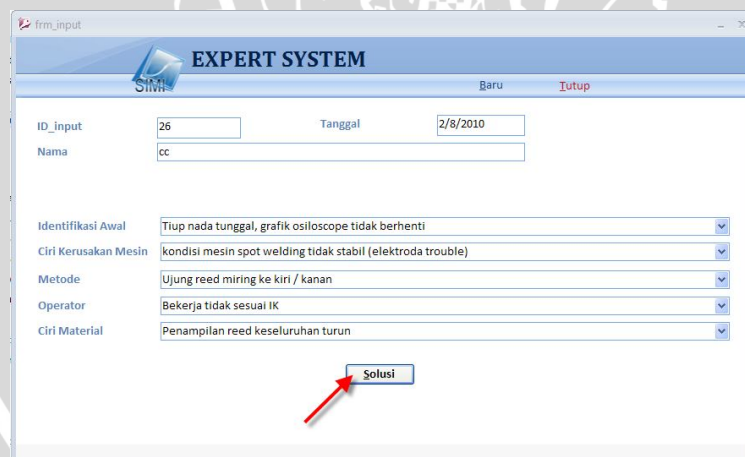
The screenshot shows a window titled 'frm_input' with the 'EXPERT SYSTEM' header. The form contains the following fields and dropdowns:

ID_input	21	Tanggal	2/1/2010
Nama	LALA		
Identifikasi Awal	Jika ditiup, pianika tidak dapat mengeluarkan bunyi sama sekali		
Ciri Kerusakan Mesin	kondisi mesin spot welding tidak stabil (parameter setting)		
Metode	Bentuk Agari Bergelombang		
Operator	Kurang kontrol / ceroboh dalam bekerja		
Ciri Material	Penampilan reed keseluruhan turun		

A red arrow points to the 'Baru' button in the top right corner of the form.

Gambar 1.27 Klik Tombol Baru

- Masukkan data tanggal, nama, identifikasi awal, kerusakan mesin, kesalahan metode, kesalahan operator, dan kondisi material. Setelah data dimasukkan semua, maka secara otomatis akan muncul tombol 'Solusi'. Klik tombol tersebut untuk melihat hasilnya.



The screenshot shows the same 'frm_input' window with the following data entered:

ID_input	26	Tanggal	2/8/2010
Nama	cc		
Identifikasi Awal	Tiup nada tunggal, grafik osiloscope tidak berhenti		
Ciri Kerusakan Mesin	kondisi mesin spot welding tidak stabil (elektroda trouble)		
Metode	Ujung reed miring ke kiri / kanan		
Operator	Bekerja tidak sesuai IK		
Ciri Material	Penampilan reed keseluruhan turun		

A red arrow points to the 'Solusi' button at the bottom center of the form.

Gambar 1.28 Klik Tombol Solusi

- Sebagai administrator, selayaknya untuk mengolah data-data dalam *expert system*, seperti data identifikasi awal, kerusakan mesin, kesalahan metode, kesalahan operator, kondisi material, jenis *defect*, proses kritis, solusi, dan *decision*. Apabila terdapat jenis *defect* baru, maka operator dapat segera memasukkan datanya melalui *Master Data Expert System*. Menu tersebut

dapat diakses melalui *main menu* dan hanya admin yang dapat mengaksesnya. Klik tombol tersebut, maka akan muncul 3 pilihan menu baru pada Master Data Expert System.

- Menu Master Data, berfungsi untuk mengolah data identifikasi awal, kerusakan mesin, kesalahan metode, kesalahan operator, kondisi material, jenis *defect*, dan proses kritis.
- Menu Solusi, berfungsi untuk mengolah data solusi cacat produk.
- Menu *Decision*, berfungsi untuk mengolah aturan-aturan *expert system*.



Gambar 1.29 Tombol *Master Data Expert System*

ID_Identifikasi	IdentifikasiAwal
0000	Tidak ada ciri defect (cacat)
0001	Tiup Dua Nada Sama, grafik osiloscope tida
0002	Tiup nada tunggal, grafik osiloscope tidak b
0003	Jika ditiup dengan tekanan tinggi atau ditiu
0004	Jika ditiup dengan tekanan rendah atau dit
0005	Bunyi terdengar berat / tersumbat
0006	Jika ditiup, pianika tidak dapat mengeluark
*	

Gambar 1.30 Form Master Data Expert System

- Form CREDIT, berfungsi untuk melihat perancang aplikasi.

Program aplikasi SIPKES ini didedikasikan untuk PT. Yamaha Musical Products Indonesia

CREDIT

Programmer and Design by:
ANGGUN PUSPITA ASRI
051067008

email: clarissa_sweety@yahoo.com
 website: http://anggunpus.deviantart.com

INDUSTRIAL ENGINEERING
BRAWIJAYA UNIVERSITY

Gambar 1.31 Form Credit

Lampiran 2: Kode Program1. Kode Program *Form SQC (form BuatChart)*

```

Option Compare Database
Private Sub btnChart32_Click()
On Error GoTo Err_Command10_Click

Dim stDocName As String
Dim buka As String
Dim tutup As String
Dim stLinkCriteria As String
Dim stDoc As String
    stDocName = "qry_P32ChartBulan"
DoCmd.OpenQuery stDocName, acNormal, acEdit
buka = "P32ChartBulan"
DoCmd.OpenForm buka, , , stLinkCriteria
    Me.Visible = False
Exit_Command10_Click:
    Exit Sub
Err_Command10_Click:
    MsgBox Err.Description
    Resume Exit_Command10_Click
End Sub

Private Sub btnChart37_Click()
On Error GoTo Err_btnChart37_Click

Dim stDocName As String
Dim buka As String
Dim tutup As String
Dim stLinkCriteria As String
Dim stDoc As String
    stDocName = "qry_P37ChartBulan"
DoCmd.OpenQuery stDocName, acNormal, acEdit
buka = "P37ChartBulan"
DoCmd.OpenForm buka, , , stLinkCriteria

```

```
Me.Visible = False
Exit_btnChart37_Click:
Exit Sub
Err_btnChart37_Click:
MsgBox Err.Description
Resume Exit_btnChart37_Click
End Sub

Private Sub btnKeluar_Click()
On Error GoTo Err_btnKeluar_Click
DoCmd.Close
Exit_btnKeluar_Click:
Exit Sub
Err_btnKeluar_Click:
MsgBox Err.Description
Resume Exit_btnKeluar_Click
End Sub

Private Sub btnChart25_Click()
On Error GoTo Err_Command10_Click
Dim stDocName As String
Dim buka As String
Dim tutup As String
Dim stLinkCriteria As String
Dim stDoc As String
stDocName = "qry_P25ChartBulan"
DoCmd.OpenQuery stDocName, acNormal, acEdit
buka = "P25ChartBulan"
DoCmd.OpenForm buka, , , stLinkCriteria
Me.Visible = False
Exit_Command10_Click:
Exit Sub
Err_Command10_Click:
MsgBox Err.Description
```

```
Resume Exit_Command10_Click
End Sub

Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
Exit Sub
Err_btnMainMenu_Click:
MsgBox Err.Description
Resume Exit_btnMainMenu_Click
End Sub

Private Sub btnPareto25_Click()
On Error GoTo Err_btnPareto25_Click

Dim stDocName As String
Dim buka As String
Dim tutup As String
Dim stLinkCriteria As String
Dim stDoc As String
stDocName = "qry_P25pareto"
DoCmd.OpenQuery stDocName, acNormal, acEdit
buka = "P25Pareto"
DoCmd.OpenForm buka, , , stLinkCriteria
Me.Visible = False
Exit_btnPareto25_Click:
Exit Sub
Err_btnPareto25_Click:
MsgBox Err.Description
Resume Exit_btnPareto25_Click
End Sub
```

```
Private Sub btnPareto32_Click()
On Error GoTo Err_btnPareto32_Click
    Dim stDocName As String
    Dim buka As String
    Dim tutup As String
    Dim stLinkCriteria As String
    Dim stDoc As String
    stDocName = "qry_P32pareto"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    buka = "P32Pareto"
    DoCmd.OpenForm buka, , , stLinkCriteria
    Me.Visible = False
Exit_btnPareto32_Click:
    Exit Sub
Err_btnPareto32_Click:
    MsgBox Err.Description
    Resume Exit_btnPareto32_Click
End Sub

Private Sub btnPareto37_Click()
On Error GoTo Err_btnPareto37_Click
    Dim stDocName As String
    Dim buka As String
    Dim tutup As String
    Dim stLinkCriteria As String
    Dim stDoc As String
    stDocName = "qry_P37pareto"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    buka = "P37Pareto"
    DoCmd.OpenForm buka, , , stLinkCriteria
    Me.Visible = False
Exit_btnPareto37_Click:
    Exit Sub
```

```
Err_btnPareto37_Click:
    MsgBox Err.Description
    Resume Exit_btnPareto37_Click
End Sub
```

2. Kode Program *Form* Laporan (*form* BuatLaporan)

```
Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Switchboard"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
    Exit Sub
Err_btnMainMenu_Click:
    MsgBox Err.Description
    Resume Exit_btnMainMenu_Click
End Sub

Private Sub btnPrintBulan25_Click()
On Error GoTo Err_btnPrintBulan25_Click
    Dim stDocName As String
    stDocName = "laporan_P25Bulan"
    DoCmd.OpenReport stDocName, acNormal
Exit_btnPrintBulan25_Click:
    Exit Sub
Err_btnPrintBulan25_Click:
    MsgBox Err.Description
    Resume Exit_btnPrintBulan25_Click
End Sub

Private Sub btnPrintBulan37_Click()
On Error GoTo Err_btnPrintBulan37_Click
```

```
Dim stDocName As String
stDocName = "laporan_P37Bulan"
DoCmd.OpenReport stDocName, acNormal
Exit_btnPrintBulan37_Click:
Exit Sub
Err_btnPrintBulan37_Click:
MsgBox Err.Description
Resume Exit_btnPrintBulan37_Click
End Sub
```

```
Private Sub btnPrintTahun25_Click()
On Error GoTo Err_btnPrintTahun25_Click
Dim stDocName As String
stDocName = "laporan_P25Tahun"
DoCmd.OpenReport stDocName, acNormal
Exit_btnPrintTahun25_Click:
Exit Sub
Err_btnPrintTahun25_Click:
MsgBox Err.Description
Resume Exit_btnPrintTahun25_Click
End Sub
```

```
Private Sub btnPrintTahun37_Click()
On Error GoTo Err_btnPrintTahun37_Click
Dim stDocName As String
stDocName = "laporan_P37Tahun"
DoCmd.OpenReport stDocName, acNormal
Exit_btnPrintTahun37_Click:
Exit Sub
Err_btnPrintTahun37_Click:
MsgBox Err.Description
Resume Exit_btnPrintTahun37_Click
End Sub
```

```
Private Sub btnReportBulan_Click()
```

```
On Error GoTo Err_btnReportBulan_Click
    Dim stDocName As String
    stDocName = "laporan_P32Bulan"
    DoCmd.OpenReport stDocName, acPreview
Exit_btnReportBulan_Click:
    Exit Sub
Err_btnReportBulan_Click:
    MsgBox Err.Description
    Resume Exit_btnReportBulan_Click
End Sub
```

```
Private Sub btnReportBulan25_Click()
On Error GoTo Err_btnReportBulan25_Click
    Dim stDocName As String
    stDocName = "laporan_P25Bulan"
    DoCmd.OpenReport stDocName, acPreview
Exit_btnReportBulan25_Click:
    Exit Sub
Err_btnReportBulan25_Click:
    MsgBox Err.Description
    Resume Exit_btnReportBulan25_Click
End Sub
```

```
Private Sub btnReportBulan37_Click()
On Error GoTo Err_btnReportBulan37_Click
    Dim stDocName As String
    stDocName = "laporan_P37Bulan"
    DoCmd.OpenReport stDocName, acPreview
Exit_btnReportBulan37_Click:
    Exit Sub
Err_btnReportBulan37_Click:
    MsgBox Err.Description
    Resume Exit_btnReportBulan37_Click
End Sub
```

```
Private Sub btnReportTahun_Click()  
On Error GoTo Err_btnReportTahun_Click  
    Dim stDocName As String  
    stDocName = "laporan_P32Tahun"  
    DoCmd.OpenReport stDocName, acPreview  
Exit_btnReportTahun_Click:  
    Exit Sub  
Err_btnReportTahun_Click:  
    MsgBox Err.Description  
    Resume Exit_btnReportTahun_Click  
End Sub
```

```
Private Sub btnPrintBulan_Click()  
On Error GoTo Err_btnPrintBulan_Click  
    Dim stDocName As String  
    stDocName = "laporan_P32Bulan"  
    DoCmd.OpenReport stDocName, acNormal  
Exit_btnPrintBulan_Click:  
    Exit Sub  
Err_btnPrintBulan_Click:  
    MsgBox Err.Description  
    Resume Exit_btnPrintBulan_Click  
End Sub
```

```
Private Sub btnPrintTahun_Click()  
On Error GoTo Err_btnPrintTahun_Click  
    Dim stDocName As String  
    stDocName = "laporan_P32Tahun"  
    DoCmd.OpenReport stDocName, acNormal  
Exit_btnPrintTahun_Click:  
    Exit Sub  
Err_btnPrintTahun_Click:  
    MsgBox Err.Description
```



```
Resume Exit_btnPrintTahun_Click
```

```
End Sub
```

```
Private Sub btnKeluar_Click()
```

```
On Error GoTo Err_btnKeluar_Click
```

```
DoCmd.Close
```

```
Exit_btnKeluar_Click:
```

```
Exit Sub
```

```
Err_btnKeluar_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnKeluar_Click
```

```
End Sub
```

```
Private Sub btnReportTahun25_Click()
```

```
On Error GoTo Err_btnReportTahun25_Click
```

```
Dim stDocName As String
```

```
stDocName = "laporan_P25Tahun"
```

```
DoCmd.OpenReport stDocName, acPreview
```

```
Exit_btnReportTahun25_Click:
```

```
Exit Sub
```

```
Err_btnReportTahun25_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnReportTahun25_Click
```

```
End Sub
```

```
Private Sub btnReportTahun37_Click()
```

```
On Error GoTo Err_btnReportTahun37_Click
```

```
Dim stDocName As String
```

```
stDocName = "laporan_P37Tahun"
```

```
DoCmd.OpenReport stDocName, acPreview
```

```
Exit_btnReportTahun37_Click:
```

```
Exit Sub
```

```
Err_btnReportTahun37_Click:
```

```
MsgBox Err.Description
Resume Exit_btnReportTahun37_Click
End Sub
```

3. Kode Program *Form* Decision

```
Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
Exit Sub
Err_btnMainMenu_Click:
MsgBox Err.Description
Resume Exit_btnMainMenu_Click
End Sub

Private Sub ComboID_Change()
ComboID_Click
End Sub

Private Sub ComboID_Click()
Dim db As New ADODB.Connection
Dim recItemP As ADODB.Recordset
Dim strSQL As String
'On Error GoTo ErrorHandler
Set db = CurrentProject.Connection
' txt.Open buka, cc, adOpenKeyset, adLockPessimistic, adCmdText
'db.Open
strSQL = "select * from tbl_solusi where ID_solusi='" & ComboID.Text & "'"
Set recItemP = db.Execute(strSQL)
If Not recItemP.EOF Then
```

```
Solusi.Value = recItemP!NamaSolusi
Else
    Solusi.Value = ""
End If
'ErrorHandlerExit:
' Exit Sub
'ErrorHandler:
' MsgBox "Error No: " & Err.Number _
' & "; Description: " & Err.Description
'Resume ErrorHandlerExit
End Sub

Private Sub btnbaru_Click()
On Error GoTo Err_btnbaru_Click
    DoCmd.GoToRecord , , acNewRec
Exit_btnbaru_Click:
    Exit Sub
Err_btnbaru_Click:
    MsgBox Err.Description
    Resume Exit_btnbaru_Click
End Sub

Private Sub btnSimpan_Click()
On Error GoTo Err_btnSimpan_Click
    DoCmd.RunCommand acCmdSaveRecord
Exit_btnSimpan_Click:
    Exit Sub
Err_btnSimpan_Click:
    MsgBox Err.Description
    Resume Exit_btnSimpan_Click
End Sub

Private Sub btnHapus_Click()
On Error GoTo Err_btnHapus_Click
```

```
DoCmd.RunCommand acCmdSelectRecord
DoCmd.RunCommand acCmdDeleteRecord
Exit_btnHapus_Click:
Exit Sub
Err_btnHapus_Click:
MsgBox Err.Description
Resume Exit_btnHapus_Click
End Sub
```

```
Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
If Me.Dirty Then Me.Dirty = False
DoCmd.Close
Exit_btnTutup_Click:
Exit Sub
Err_btnTutup_Click:
MsgBox Err.Description
Resume Exit_btnTutup_Click
End Sub
```

```
Private Sub btnRule_Click()
On Error GoTo Err_btnRule_Click
Dim stDocName As String
stDocName = "RuleSet"
DoCmd.OpenReport stDocName, acPreview
Exit_btnRule_Click:
Exit Sub
Err_btnRule_Click:
MsgBox Err.Description
Resume Exit_btnRule_Click
End Sub
```

4. Kode Program *Form* Defects

```
Option Compare Database
Private Sub btnbaru_Click()
On Error GoTo Err_btnbaru_Click
    DoCmd.GoToRecord , , acNewRec
Exit_btnbaru_Click:
    Exit Sub
Err_btnbaru_Click:
    MsgBox Err.Description
    Resume Exit_btnbaru_Click
End Sub
```

```
Private Sub btnSimpan_Click()
On Error GoTo Err_btnSimpan_Click
    DoCmd.RunCommand acCmdSaveRecord
Exit_btnSimpan_Click:
    Exit Sub
Err_btnSimpan_Click:
    MsgBox Err.Description
    Resume Exit_btnSimpan_Click
End Sub
```

```
Private Sub btnHapus_Click()
On Error GoTo Err_btnHapus_Click
    DoCmd.RunCommand acCmdSelectRecord
    DoCmd.RunCommand acCmdDeleteRecord
Exit_btnHapus_Click:
    Exit Sub
Err_btnHapus_Click:
    MsgBox Err.Description
    Resume Exit_btnHapus_Click
End Sub
```

```
Private Sub btnPrev_Click()
On Error GoTo Err_btnPrev_Click
    DoCmd.GoToRecord , , acPrevious
```

```
Exit_btnPrev_Click:
    Exit Sub
Err_btnPrev_Click:
    MsgBox Err.Description
    Resume Exit_btnPrev_Click
End Sub
```

```
Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
DoCmd.GoToRecord , , acNext
```

```
Exit_btnNext_Click:
    Exit Sub
Err_btnNext_Click:
    MsgBox Err.Description
    Resume Exit_btnNext_Click
End Sub
```

```
Private Sub Combo6_AfterUpdate()
    Dim rs As Object
    Set rs = Me.Recordset.Clone
    rs.FindFirst "[IDDefect] = " & Me![Combo6] & ""
    If Not rs.EOF Then Me.Bookmark = rs.Bookmark
End Sub
```

5. Kode Program *Form* Input Expert System (*form frm_input*)

```
Dim p1Val, p2Val, p3Val, p4Val, p5Val, inputID, namaInput, tanggalInput,
lengkapi
```

```
Private Sub btnSolusi_Click()
```

```
'inference engine dengan strSql atau recordset, UPDATE TABEL
```

```
Dim cn As ADODB.Connection
```

```
Dim rs1 As New ADODB.Recordset
```

```
Dim rs2 As New ADODB.Recordset
```

```
Dim rs3 As New ADODB.Recordset
```

```

Dim strSql1, strSql2, strSql3 As String
    Set cn = CurrentProject.Connection
    strSql1 = "select * from tbl_input"
rs1.Open strSql1, cn, adOpenKeyset, adLockPessimistic, adCmdText
'Do While Not rs1.EOF 'jelajahi tbl_input sebagai input
cn.Execute "update tbl_decision set fire=null where fire=1", , adCmdText
cn.Execute "delete * from tbl_sementara", , adCmdText
    strSql2 = "select * from tbl_decision"
rs2.Open strSql2, cn, adOpenKeyset, adLockPessimistic, adCmdText
rs2.MoveFirst
Do While Not rs2.EOF 'jelajahi tabel Decision
    If (rs2!P1 = p1Val) And (rs2!P2 = p2Val) And (rs2!P3 = p3Val) And
(rs2!P4 = p4Val) And (rs2!P5 = p5Val) Then
        rs2!Fire = 1 'Rule meledak
        rs2.Update
    End If
    rs2.MoveNext
Loop
rs2.MoveFirst
If namaInput = "" Or tanggalInput = "" Then
lengkapi = MsgBox("Lengkapi Data Nama user dan tanggal", vbOKOnly)
Exit Sub
End If
Do While Not rs2.EOF
If rs2!Fire = 1 Then
Set rs3 = New ADODB.Recordset
With rs3
.Open "tbl_sementara", CurrentProject.Connection, adOpenDynamic, _
adLockPessimistic
.AddNew
.Fields("ID_input") = inputID
.Fields("Nama") = namaInput
.Fields("Tanggal") = tanggalInput
.Fields("P1") = rs2!P1

```

```
.Fields("P2") = rs2!P2
.Fields("P3") = rs2!P3
.Fields("P4") = rs2!P4
.Fields("P5") = rs2!P5
.Fields("S1") = rs2!S1
.Fields("S2") = rs2!S2
.Fields("Solusi") = rs2!Solusi
.Update
End With
Set rs3 = Nothing
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "rpt_hasil"
DoCmd.OpenReport stDocName, acPreview
End If
rs2.MoveNext
Loop
rs2.Close
rs1.Close
End Sub

Private Sub ID_input_BeforeUpdate(Cancel As Integer)
inputID = ID_input.Text
End Sub

Private Sub Nama_AfterUpdate()
namaInput = Nama.Text
End Sub

Private Sub Tanggal_AfterUpdate()
tanggalInput = Tanggal.Text
End Sub

Private Sub P1_Change()
```

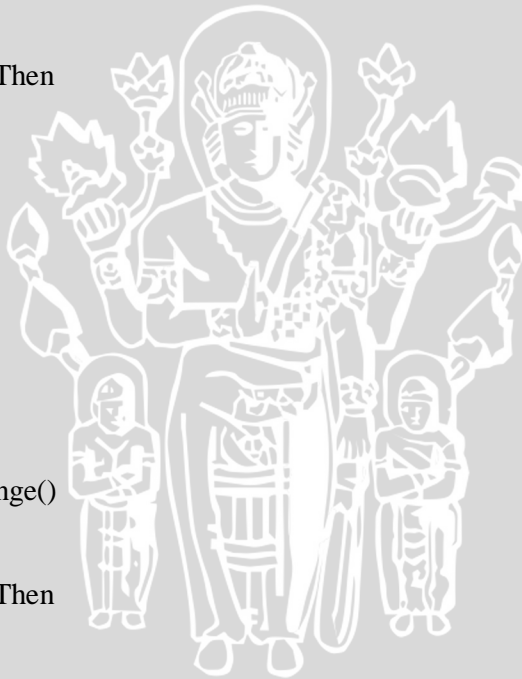


```
p1Val = P1.Text
If P1.Text = Empty Then
    p1Val = Null
End If
P2.Enabled = True
p2Val = Null
p3Val = Null
p4Val = Null
p5Val = Null
End Sub
```

```
Private Sub P2_Change()
p2Val = P2.Text
If P2.Text = Empty Then
    p2Val = Null
End If
P3.Enabled = True
p3Val = Null
p4Val = Null
End Sub
```

```
Private Sub P3_Change()
p3Val = P3.Text
If P3.Text = Empty Then
    p3Val = Null
End If
P4.Enabled = True
p4Val = Null
End Sub
```

```
Private Sub P4_Change()
p4Val = P4.Text
If P4.Text = Empty Then
    p4Val = Null
End If
```



```
P5.Enabled = True
p5Val = Null
End Sub
```

```
Private Sub P5_Change()
p5Val = P5.Text
If P5.Text = Empty Then
    p5Val = Null
End If
btnSolusi.Enabled = True
End Sub
```

```
Private Sub btnbaru_Click()
On Error GoTo Err_btnbaru_Click
DoCmd.GoToRecord , , acNewRec
namaInput = ""
tanggalInput = ""
btnSolusi.Enabled = False
P2.Enabled = False
P3.Enabled = False
P4.Enabled = False
P5.Enabled = False
Exit_btnbaru_Click:
Exit Sub
Err_btnbaru_Click:
MsgBox Err.Description
Resume Exit_btnbaru_Click
End Sub
```

```
Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
If Me.Dirty Then Me.Dirty = False
DoCmd.Close
Exit_btnTutup_Click:
```

```

Exit Sub
Err_btnTutup_Click:
    MsgBox Err.Description
Resume Exit_btnTutup_Click
End Sub

```

6. Kode Program *Form* Graphs

```

Option Compare Database
Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
    DoCmd.Close
Exit_btnTutup_Click:
    Exit Sub
Err_btnTutup_Click:
    MsgBox Err.Description
Resume Exit_btnTutup_Click
End Sub

Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Switchboard"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
    Exit Sub
Err_btnMainMenu_Click:
    MsgBox Err.Description
Resume Exit_btnMainMenu_Click
End Sub

```

7. Kode Program *Form* Identifikasi Awal

```

Option Compare Database

```

```
Private Sub btnbaru_Click()  
On Error GoTo Err_btnbaru_Click  
    DoCmd.GoToRecord , , acNewRec  
Exit_btnbaru_Click:  
    Exit Sub  
Err_btnbaru_Click:  
    MsgBox Err.Description  
    Resume Exit_btnbaru_Click  
End Sub
```

```
Private Sub btnSimpan_Click()  
On Error GoTo Err_btnSimpan_Click  
    DoCmd.RunCommand acCmdSaveRecord  
Exit_btnSimpan_Click:  
    Exit Sub  
Err_btnSimpan_Click:  
    MsgBox Err.Description  
    Resume Exit_btnSimpan_Click  
End Sub
```

```
Private Sub btnHapus_Click()  
On Error GoTo Err_btnHapus_Click  
    DoCmd.RunCommand acCmdSelectRecord  
    DoCmd.RunCommand acCmdDeleteRecord  
Exit_btnHapus_Click:  
    Exit Sub  
Err_btnHapus_Click:  
    MsgBox Err.Description  
    Resume Exit_btnHapus_Click  
End Sub
```

```
Private Sub btnPrev_Click()  
On Error GoTo Err_btnPrev_Click  
    DoCmd.GoToRecord , , acPrevious
```

```
Exit_btnPrev_Click:
    Exit Sub
Err_btnPrev_Click:
    MsgBox Err.Description
    Resume Exit_btnPrev_Click
End Sub
```

```
Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
    DoCmd.GoToRecord , , acNext
```

```
Exit_btnNext_Click:
    Exit Sub
Err_btnNext_Click:
    MsgBox Err.Description
    Resume Exit_btnNext_Click
End Sub
```

8. Kode Program *Form* Master Data

```
Option Compare Database
Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
    DoCmd.Close
Exit_btnTutup_Click:
    Exit Sub
Err_btnTutup_Click:
    MsgBox Err.Description
    Resume Exit_btnTutup_Click
End Sub

Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
```

```
stDocName = "Switchboard"  
DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_btnMainMenu_Click:  
Exit Sub  
Err_btnMainMenu_Click:  
MsgBox Err.Description  
Resume Exit_btnMainMenu_Click  
End Sub
```

9. Kode Program *Form* Material

```
Option Compare Database  
Private Sub btnbaru_Click()  
On Error GoTo Err_btnbaru_Click  
DoCmd.GoToRecord , , acNewRec  
Exit_btnbaru_Click:  
Exit Sub  
Err_btnbaru_Click:  
MsgBox Err.Description  
Resume Exit_btnbaru_Click  
End Sub  
  
Private Sub btnSimpan_Click()  
On Error GoTo Err_btnSimpan_Click  
DoCmd.RunCommand acCmdSaveRecord  
Exit_btnSimpan_Click:  
Exit Sub  
Err_btnSimpan_Click:  
MsgBox Err.Description  
Resume Exit_btnSimpan_Click  
End Sub  
  
Private Sub btnHapus_Click()  
On Error GoTo Err_btnHapus_Click
```

```
DoCmd.RunCommand acCmdSelectRecord
DoCmd.RunCommand acCmdDeleteRecord
Exit_btnHapus_Click:
Exit Sub
Err_btnHapus_Click:
MsgBox Err.Description
Resume Exit_btnHapus_Click
End Sub
```

```
Private Sub btnPrev_Click()
On Error GoTo Err_btnPrev_Click
DoCmd.GoToRecord , , acPrevious
Exit_btnPrev_Click:
Exit Sub
```

```
Err_btnPrev_Click:
MsgBox Err.Description
Resume Exit_btnPrev_Click
End Sub
```

```
Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
Exit Sub
```

```
Err_btnNext_Click:
MsgBox Err.Description
Resume Exit_btnNext_Click
End Sub
```

10. Kode Program *Form* Mesin

```
Option Compare Database
Private Sub btnbaru_Click()
```

```
On Error GoTo Err_btnbaru_Click  
DoCmd.GoToRecord , , acNewRec
```

```
Exit_btnbaru_Click:
```

```
Exit Sub
```

```
Err_btnbaru_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnbaru_Click
```

```
End Sub
```

```
Private Sub btnSimpan_Click()
```

```
On Error GoTo Err_btnSimpan_Click
```

```
DoCmd.RunCommand acCmdSaveRecord
```

```
Exit_btnSimpan_Click:
```

```
Exit Sub
```

```
Err_btnSimpan_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnSimpan_Click
```

```
End Sub
```

```
Private Sub btnHapus_Click()
```

```
On Error GoTo Err_btnHapus_Click
```

```
DoCmd.RunCommand acCmdSelectRecord
```

```
DoCmd.RunCommand acCmdDeleteRecord
```

```
Exit_btnHapus_Click:
```

```
Exit Sub
```

```
Err_btnHapus_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnHapus_Click
```

```
End Sub
```

```
Private Sub btnPrev_Click()
```

```
On Error GoTo Err_btnPrev_Click
```

```
DoCmd.GoToRecord , , acPrevious
```

```
Exit_btnPrev_Click:
```



```
Exit Sub
Err_btnPrev_Click:
MsgBox Err.Description
Resume Exit_btnPrev_Click
End Sub

Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
Exit Sub
Err_btnNext_Click:
MsgBox Err.Description
Resume Exit_btnNext_Click
End Sub
```

11. Kode Program *Form* Metode

```
Option Compare Database
Private Sub btnbaru_Click()
On Error GoTo Err_btnbaru_Click
DoCmd.GoToRecord , , acNewRec
Exit_btnbaru_Click:
Exit Sub
Err_btnbaru_Click:
MsgBox Err.Description
Resume Exit_btnbaru_Click
End Sub

Private Sub btnSimpan_Click()
On Error GoTo Err_btnSimpan_Click
DoCmd.RunCommand acCmdSaveRecord
Exit_btnSimpan_Click:
Exit Sub
```

```
Err_btnSimpan_Click:
    MsgBox Err.Description
    Resume Exit_btnSimpan_Click
End Sub

Private Sub btnHapus_Click()
On Error GoTo Err_btnHapus_Click
    DoCmd.RunCommand acCmdSelectRecord
    DoCmd.RunCommand acCmdDeleteRecord
Exit_btnHapus_Click:
    Exit Sub
Err_btnHapus_Click:
    MsgBox Err.Description
    Resume Exit_btnHapus_Click
End Sub

Private Sub btnPrev_Click()
On Error GoTo Err_btnPrev_Click
    DoCmd.GoToRecord , , acPrevious
Exit_btnPrev_Click:
    Exit Sub
Err_btnPrev_Click:
    MsgBox Err.Description
    Resume Exit_btnPrev_Click
End Sub

Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
    DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
    Exit Sub
Err_btnNext_Click:
    MsgBox Err.Description
    Resume Exit_btnNext_Click
End Sub
```

12. Kode Program *Form Operator*

```
Option Compare Database
Private Sub btnbaru_Click()
On Error GoTo Err_btnbaru_Click
DoCmd.GoToRecord , , acNewRec
Exit_btnbaru_Click:
Exit Sub
Err_btnbaru_Click:
MsgBox Err.Description
Resume Exit_btnbaru_Click
End Sub

Private Sub btnSimpan_Click()
On Error GoTo Err_btnSimpan_Click
DoCmd.RunCommand acCmdSaveRecord
Exit_btnSimpan_Click:
Exit Sub
Err_btnSimpan_Click:
MsgBox Err.Description
Resume Exit_btnSimpan_Click
End Sub

Private Sub btnHapus_Click()
On Error GoTo Err_btnHapus_Click
DoCmd.RunCommand acCmdSelectRecord
DoCmd.RunCommand acCmdDeleteRecord
Exit_btnHapus_Click:
Exit Sub
Err_btnHapus_Click:
MsgBox Err.Description
Resume Exit_btnHapus_Click
End Sub
```

```

Private Sub btnPrev_Click()
On Error GoTo Err_btnPrev_Click
    DoCmd.GoToRecord , , acPrevious
Exit_btnPrev_Click:
    Exit Sub
Err_btnPrev_Click:
    MsgBox Err.Description
    Resume Exit_btnPrev_Click
End Sub

```

```

Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
    DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
    Exit Sub
Err_btnNext_Click:
    MsgBox Err.Description
    Resume Exit_btnNext_Click
End Sub

```

13. Kode Program *Form* P25

```

Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Switchboard"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
    Exit Sub
Err_btnMainMenu_Click:
    MsgBox Err.Description

```

```
Resume Exit_btnMainMenu_Click
End Sub

Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
    DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
    Exit Sub
Err_btnNext_Click:
    MsgBox Err.Description
Resume Exit_btnNext_Click
End Sub

Private Sub btnPrev_Click()
On Error GoTo Err_btnPrev_Click
    DoCmd.GoToRecord , , acPrevious
Exit_btnPrev_Click:
    Exit Sub
Err_btnPrev_Click:
    MsgBox Err.Description
Resume Exit_btnPrev_Click
End Sub

Private Sub btnPetaKontrol_Click()
On Error GoTo Err_btnPetaKontrol_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "BuatChart"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnPetaKontrol_Click:
    Exit Sub
Err_btnPetaKontrol_Click:
    MsgBox Err.Description
```

```
Resume Exit_btnPetaKontrol_Click
End Sub

Private Sub btnGraphs_Click()
On Error GoTo Err_btnHistogram_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Graphs"
DoCmd.OpenForm stDocName, , adModeRead
Exit_btnHistogram_Click:
Exit Sub
Err_btnHistogram_Click:
MsgBox Err.Description
Resume Exit_btnHistogram_Click
End Sub

Private Sub btnbaru_Click()
On Error GoTo Err_btnbaru_Click
DoCmd.GoToRecord , , acNewRec
Exit_btnbaru_Click:
Exit Sub
Err_btnbaru_Click:
MsgBox Err.Description
Resume Exit_btnbaru_Click
End Sub

Private Sub btnSimpan_Click()
On Error GoTo Err_btnSimpan_Click
DoCmd.RunCommand acCmdSaveRecord
Exit_btnSimpan_Click:
Exit Sub
Err_btnSimpan_Click:
MsgBox Err.Description
Resume Exit_btnSimpan_Click
```

```
End Sub
```

```
Private Sub btnHapus_Click()
```

```
On Error GoTo Err_btnHapus_Click
```

```
DoCmd.RunCommand acCmdSelectRecord
```

```
DoCmd.RunCommand acCmdDeleteRecord
```

```
Exit_btnHapus_Click:
```

```
Exit Sub
```

```
Err_btnHapus_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnHapus_Click
```

```
End Sub
```

```
Private Sub btnPrint_Click()
```

```
On Error GoTo Err_btnPrint_Click
```

```
DoCmd.PrintOut
```

```
Exit_btnPrint_Click:
```

```
Exit Sub
```

```
Err_btnPrint_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnPrint_Click
```

```
End Sub
```

```
Private Sub btnTutup_Click()
```

```
On Error GoTo Err_btnTutup_Click
```

```
If Me.Dirty Then Me.Dirty = False
```

```
DoCmd.Close
```

```
Exit_btnTutup_Click:
```

```
Exit Sub
```

```
Err_btnTutup_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnTutup_Click
```

```
End Sub
```

```
Private Sub btnFormLaporan_Click()
On Error GoTo Err_btnFormLaporan_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "BuatLaporan"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_btnFormLaporan_Click:
    Exit Sub
Err_btnFormLaporan_Click:
    MsgBox Err.Description
    Resume Exit_btnFormLaporan_Click
End Sub
```

14. Kode Program *Form* Peta Kontrol P25 (*form* P25ChartBulan)

```
Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Switchboard"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
    Exit Sub
Err_btnMainMenu_Click:
    MsgBox Err.Description
    Resume Exit_btnMainMenu_Click
End Sub
```

```
Private Sub Command8_Click()
    Dim rs As Recordset
    Dim sql As String
    Dim qcatt As QCAtrib ' Create object variable.
```



```
sql = "Select * From tbl_P25ChartBulan"
Set rs = CurrentDb.OpenRecordset(sql)
    rs.OpenRecordset
Set qcatt = QCAtrib1.Object
qcatt.CCReset
Dim i As Integer
Dim cacat As Double
Dim total As Double
Dim id As Integer
    i = 0
Do Until rs.EOF
    i = i + 1
    cacat = rs("p")
    total = rs("n")
    id = rs("ID_data")
    With qcatt
        .CCAddNonConforms (cacat)
        .CCSetSampleSize i, (total)
        .CCSetId i, (id)
    End With
    rs.MoveNext
Loop
qcatt.Refresh
rs.Close
Set rs = Nothing
End Sub

Private Sub btnES_Click()
On Error GoTo Err_btnES_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "POP_UP_ES"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnES_Click:
```

```
Exit Sub
Err_btnES_Click:
    MsgBox Err.Description
    Resume Exit_btnES_Click
End Sub

Private Sub btnBuatChart_Click()
On Error GoTo Err_btnBuatChart_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim stDoc As String
    stDocName = "BuatChart"
    DoCmd.OpenForm stDocName, , stLinkCriteria
    stDoc = "P25ChartBulan"
    DoCmd.Close acForm, stDoc
Exit_btnBuatChart_Click:
    Exit Sub
Err_btnBuatChart_Click:
    MsgBox Err.Description
    Resume Exit_btnBuatChart_Click
End Sub

Private Sub btnPrint_Click()
On Error GoTo Err_btnPrint_Click
    DoCmd.PrintOut
Exit_btnPrint_Click:
    Exit Sub
Err_btnPrint_Click:
    MsgBox Err.Description
    Resume Exit_btnPrint_Click
End Sub

Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
    If Me.Dirty Then Me.Dirty = False
```

```
DoCmd.Close
Exit_btnTutup_Click:
Exit Sub
Err_btnTutup_Click:
MsgBox Err.Description
Resume Exit_btnTutup_Click
End Sub
```

15. Kode Program *Form* P25Pareto

```
Option Compare Database
Private Sub Command1_Click()
Dim rs As Recordset
Dim sql As String
Dim pp As pareto
sql = "Select * From P25Pareto"
Set rs = CurrentDb.OpenRecordset(sql)
rs.OpenRecordset
Set pp = Pareto0.Object
pp.Refresh
Dim satu As Double
Dim dua As Double
Dim tiga As Double
Dim empat As Double
Dim lima As Double
Dim enam As Double
satu = rs("SumOfoktaf_aneh")
dua = rs("SumOfBiri")
tiga = rs("SumOfReed_tinggi")
empat = rs("SumOfReed_rendah")
lima = rs("SumOfBunyi_tersumbat")
enam = rs("SumOfTidak_bunyi")
pp.Y1AxisText = PShowCountAxis
pp.Y1AxisInterval = 10
```

```

pp.Y2AxisText = PShowPercentAxis
pp.Y2AxisInterval = 10
pp.Count(1) = satu
pp.Color(1) = vbMagenta
pp.Category(1) = "oktaf aneh"
pp.Count(2) = dua
pp.Color(2) = vbBlue
pp.Category(2) = "Biri"
pp.Count(3) = tiga
pp.Color(3) = vbGreen
pp.Category(3) = "Reed tinggi"
pp.Count(4) = empat
pp.Color(4) = vbRed
pp.Category(4) = "Reed rendah"
pp.Count(5) = lima
pp.Color(5) = vbBlack
pp.Category(5) = "Bunyi tersumbat"
pp.Count(6) = enam
pp.Color(6) = vbMagenta
pp.Category(6) = "Tidak bunyi"
pp.GraphContent = PCount
pp.Refresh
rs.Close
Set rs = Nothing
End Sub

```

```

Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
Exit Sub

```

```
Err_btnMainMenu_Click:
    MsgBox Err.Description
    Resume Exit_btnMainMenu_Click
End Sub
```

```
Private Sub btnKeluar_Click()
On Error GoTo Err_btnKeluar_Click
DoCmd.Close
```

```
Exit_btnKeluar_Click:
Exit Sub
Err_btnKeluar_Click:
MsgBox Err.Description
Resume Exit_btnKeluar_Click
```

```
End Sub
```

16. Kode Program *Form P32*

```
Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_btnMainMenu_Click:
Exit Sub
Err_btnMainMenu_Click:
MsgBox Err.Description
Resume Exit_btnMainMenu_Click
```

```
End Sub
```

```
Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
DoCmd.GoToRecord , , acNext
```

```
Exit_btnNext_Click:
```

```
Exit Sub
```

```
Err_btnNext_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnNext_Click
```

```
End Sub
```

```
Private Sub btnPrev_Click()
```

```
On Error GoTo Err_btnPrev_Click
```

```
DoCmd.GoToRecord , , acPrevious
```

```
Exit_btnPrev_Click:
```

```
Exit Sub
```

```
Err_btnPrev_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnPrev_Click
```

```
End Sub
```

```
Private Sub btnPetaKontrol_Click()
```

```
On Error GoTo Err_btnPetaKontrol_Click
```

```
Dim stDocName As String
```

```
Dim stLinkCriteria As String
```

```
stDocName = "BuatChart"
```

```
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_btnPetaKontrol_Click:
```

```
Exit Sub
```

```
Err_btnPetaKontrol_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnPetaKontrol_Click
```

```
End Sub
```

```
Private Sub btnGraphs_Click()
```

```
On Error GoTo Err_btnHistogram_Click
```

```
Dim stDocName As String
```

```
Dim stLinkCriteria As String
```

```
stDocName = "Graphs"  
DoCmd.OpenForm stDocName, , , adModeRead  
Exit_btnHistogram_Click:  
Exit Sub  
Err_btnHistogram_Click:  
MsgBox Err.Description  
Resume Exit_btnHistogram_Click  
End Sub
```

```
Private Sub btnbaru_Click()  
On Error GoTo Err_btnbaru_Click  
DoCmd.GoToRecord , , acNewRec  
Exit_btnbaru_Click:  
Exit Sub  
Err_btnbaru_Click:  
MsgBox Err.Description  
Resume Exit_btnbaru_Click  
End Sub
```

```
Private Sub btnSimpan_Click()  
On Error GoTo Err_btnSimpan_Click  
DoCmd.RunCommand acCmdSaveRecord  
Exit_btnSimpan_Click:  
Exit Sub  
Err_btnSimpan_Click:  
MsgBox Err.Description  
Resume Exit_btnSimpan_Click  
End Sub
```

```
Private Sub btnHapus_Click()  
On Error GoTo Err_btnHapus_Click  
DoCmd.RunCommand acCmdSelectRecord  
DoCmd.RunCommand acCmdDeleteRecord  
Exit_btnHapus_Click:
```

```
Exit Sub
Err_btnHapus_Click:
    MsgBox Err.Description
Resume Exit_btnHapus_Click
End Sub

Private Sub btnPrint_Click()
On Error GoTo Err_btnPrint_Click
DoCmd.PrintOut
Exit_btnPrint_Click:
Exit Sub
Err_btnPrint_Click:
MsgBox Err.Description
Resume Exit_btnPrint_Click
End Sub

Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
If Me.Dirty Then Me.Dirty = False
DoCmd.Close
Exit_btnTutup_Click:
Exit Sub
Err_btnTutup_Click:
MsgBox Err.Description
Resume Exit_btnTutup_Click
End Sub

Private Sub btnFormLaporan_Click()
On Error GoTo Err_btnFormLaporan_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "BuatLaporan"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnFormLaporan_Click:
```



```

Exit Sub
Err_btnFormLaporan_Click:
MsgBox Err.Description
Resume Exit_btnFormLaporan_Click
End Sub

```

17. Kode Program *Form* Peta Kontrol P32 (*form* P32ChartBulan)

```

Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , stLinkCriteria
Exit_btnMainMenu_Click:
Exit Sub
Err_btnMainMenu_Click:
MsgBox Err.Description
Resume Exit_btnMainMenu_Click
End Sub

Private Sub Command8_Click()
Dim rs As Recordset
Dim sql As String
Dim qcatt As QCAAttrib ' Create object variable.

sql = "Select * From tbl_P32ChartBulan"
Set rs = CurrentDb.OpenRecordset(sql)
rs.OpenRecordset
Set qcatt = QCAAttrib1.Object
qcatt.CCReset
Dim i As Integer
Dim cacat As Double

```

```
Dim total As Double
Dim id As Integer
i = 0
Do Until rs.EOF
    i = i + 1
    cacat = rs("p")
    total = rs("n")
    id = rs("ID_data")
    With qcatt
        .CCAddNonConforms (cacat)
        .CCSetSampleSize i, (total)
        .CCSetId i, (id)
    End With
    rs.MoveNext
Loop
qcatt.Refresh
rs.Close
Set rs = Nothing
End Sub

Private Sub btnES_Click()
On Error GoTo Err_btnES_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "POP_UP_ES"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnES_Click:
Exit Sub
Err_btnES_Click:
    MsgBox Err.Description
    Resume Exit_btnES_Click
End Sub

Private Sub btnBuatChart_Click()
```

```
On Error GoTo Err_btnBuatChart_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim stDoc As String
    stDocName = "P32BuatChart"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    stDoc = "P32ChartBulan"
    DoCmd.Close acForm, stDoc
Exit_btnBuatChart_Click:
    Exit Sub
Err_btnBuatChart_Click:
    MsgBox Err.Description
    Resume Exit_btnBuatChart_Click
End Sub

Private Sub btnPrint_Click()
On Error GoTo Err_btnPrint_Click
    DoCmd.PrintOut
Exit_btnPrint_Click:
    Exit Sub
Err_btnPrint_Click:
    MsgBox Err.Description
    Resume Exit_btnPrint_Click
End Sub

Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
    If Me.Dirty Then Me.Dirty = False
    DoCmd.Close
Exit_btnTutup_Click:
    Exit Sub
Err_btnTutup_Click:
    MsgBox Err.Description
    Resume Exit_btnTutup_Click
```

End Sub

18. Kode Program *Form P32Pareto*

Option Compare Database

Private Sub Command1_Click()

Dim rs As Recordset

Dim sql As String

Dim pp As pareto

sql = "Select * From P32Pareto"

Set rs = CurrentDb.OpenRecordset(sql)

rs.OpenRecordset

Set pp = Pareto0.Object

pp.Refresh

Dim satu As Double

Dim dua As Double

Dim tiga As Double

Dim empat As Double

Dim lima As Double

Dim enam As Double

satu = rs("SumOfoktaf_aneh")

dua = rs("SumOfBiri")

tiga = rs("SumOfReed_tinggi")

empat = rs("SumOfReed_rendah")

lima = rs("SumOfBunyi_tersumbat")

enam = rs("SumOfTidak_bunyi")

pp.Y1AxisText = PShowCountAxis

pp.Y1AxisInterval = 10

pp.Y2AxisText = PShowPercentAxis

pp.Y2AxisInterval = 10

pp.Count(1) = satu

pp.Color(1) = vbMagenta

pp.Category(1) = "oktaf aneh"

pp.Count(2) = dua

```
pp.Color(2) = vbBlue
pp.Category(2) = "Biri"
pp.Count(3) = tiga
pp.Color(3) = vbGreen
pp.Category(3) = "Reed tinggi"
pp.Count(4) = empat
pp.Color(4) = vbRed
pp.Category(4) = "Reed rendah"
pp.Count(5) = lima
pp.Color(5) = vbBlack
pp.Category(5) = "Bunyi tersumbat"
pp.Count(6) = enam
pp.Color(6) = vbMagenta
pp.Category(6) = "Tidak bunyi"
pp.GraphContent = PCount
pp.Refresh
rs.Close
Set rs = Nothing
End Sub

Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
Exit Sub
Err_btnMainMenu_Click:
MsgBox Err.Description
Resume Exit_btnMainMenu_Click
End Sub

Private Sub btnKeluar_Click()
```

```
On Error GoTo Err_btnKeluar_Click
    DoCmd.Close
Exit_btnKeluar_Click:
    Exit Sub
Err_btnKeluar_Click:
    MsgBox Err.Description
    Resume Exit_btnKeluar_Click
End Sub
```

19. Kode Program *Form P37*

```
Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Switchboard"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
    Exit Sub
Err_btnMainMenu_Click:
    MsgBox Err.Description
    Resume Exit_btnMainMenu_Click
End Sub

Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
    DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
    Exit Sub
Err_btnNext_Click:
    MsgBox Err.Description
    Resume Exit_btnNext_Click
End Sub
```

```
Private Sub btnPrev_Click()  
On Error GoTo Err_btnPrev_Click  
    DoCmd.GoToRecord , , acPrevious  
Exit_btnPrev_Click:  
    Exit Sub  
Err_btnPrev_Click:  
    MsgBox Err.Description  
    Resume Exit_btnPrev_Click  
End Sub
```

```
Private Sub btnPetaKontrol_Click()  
On Error GoTo Err_btnPetaKontrol_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "BuatChart"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_btnPetaKontrol_Click:  
    Exit Sub  
Err_btnPetaKontrol_Click:  
    MsgBox Err.Description  
    Resume Exit_btnPetaKontrol_Click  
End Sub
```

```
Private Sub btnGraphs_Click()  
On Error GoTo Err_btnHistogram_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "Graphs"  
    DoCmd.OpenForm stDocName, , , adModeRead  
Exit_btnHistogram_Click:  
    Exit Sub  
Err_btnHistogram_Click:  
    MsgBox Err.Description
```

```
Resume Exit_btnHistogram_Click
```

```
End Sub
```

```
Private Sub btnbaru_Click()
```

```
On Error GoTo Err_btnbaru_Click
```

```
DoCmd.GoToRecord , , acNewRec
```

```
Exit_btnbaru_Click:
```

```
Exit Sub
```

```
Err_btnbaru_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnbaru_Click
```

```
End Sub
```

```
Private Sub btnSimpan_Click()
```

```
On Error GoTo Err_btnSimpan_Click
```

```
DoCmd.RunCommand acCmdSaveRecord
```

```
Exit_btnSimpan_Click:
```

```
Exit Sub
```

```
Err_btnSimpan_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnSimpan_Click
```

```
End Sub
```

```
Private Sub btnHapus_Click()
```

```
On Error GoTo Err_btnHapus_Click
```

```
DoCmd.RunCommand acCmdSelectRecord
```

```
DoCmd.RunCommand acCmdDeleteRecord
```

```
Exit_btnHapus_Click:
```

```
Exit Sub
```

```
Err_btnHapus_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_btnHapus_Click
```

```
End Sub
```



```
Private Sub btnPrint_Click()  
On Error GoTo Err_btnPrint_Click  
    DoCmd.PrintOut  
Exit_btnPrint_Click:  
    Exit Sub  
Err_btnPrint_Click:  
    MsgBox Err.Description  
    Resume Exit_btnPrint_Click  
End Sub  
  
Private Sub btnTutup_Click()  
On Error GoTo Err_btnTutup_Click  
    If Me.Dirty Then Me.Dirty = False  
    DoCmd.Close  
Exit_btnTutup_Click:  
    Exit Sub  
Err_btnTutup_Click:  
    MsgBox Err.Description  
    Resume Exit_btnTutup_Click  
End Sub  
  
Private Sub btnFormLaporan_Click()  
On Error GoTo Err_btnFormLaporan_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "BuatLaporan"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_btnFormLaporan_Click:  
    Exit Sub  
Err_btnFormLaporan_Click:  
    MsgBox Err.Description  
    Resume Exit_btnFormLaporan_Click  
End Sub
```

20. Kode Program *Form* Peta Kontrol P37 (*form* P37ChartBulan)

```
Option Compare Database
Private Sub btnMainMenu_Click()
On Error GoTo Err_btnMainMenu_Click

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnMainMenu_Click:
Exit Sub
Err_btnMainMenu_Click:
MsgBox Err.Description
Resume Exit_btnMainMenu_Click
End Sub

Private Sub Command8_Click()
Dim rs As Recordset
Dim sql As String
Dim qcatt As QCAattrib ' Create object variable.
sql = "Select * From tbl_P37ChartBulan"
Set rs = CurrentDb.OpenRecordset(sql)
rs.OpenRecordset
Set qcatt = QCAattrib1.Object
qcatt.CCReset
Dim i As Integer
Dim cacat As Double
Dim total As Double
Dim id As Integer
i = 0
Do Until rs.EOF
i = i + 1
cacat = rs("p")
total = rs("n")
```

```
id = rs("ID_data")
With qcatt
    .CCAddNonConforms (cacat)
    .CCSetSampleSize i, (total)
    .CCSetId i, (id)
End With
rs.MoveNext

Loop
qcatt.Refresh
rs.Close
Set rs = Nothing
End Sub

Private Sub btnES_Click()
On Error GoTo Err_btnES_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "POP_UP_ES"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_btnES_Click:
    Exit Sub
Err_btnES_Click:
    MsgBox Err.Description
    Resume Exit_btnES_Click
End Sub

Private Sub btnBuatChart_Click()
On Error GoTo Err_btnBuatChart_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim stDoc As String
    stDocName = "BuatChart"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    stDoc = "P37ChartBulan"
```

```

DoCmd.Close acForm, stDoc
Exit_btnBuatChart_Click:
Exit Sub
Err_btnBuatChart_Click:
MsgBox Err.Description
Resume Exit_btnBuatChart_Click
End Sub

```

```

Private Sub btnPrint_Click()
On Error GoTo Err_btnPrint_Click

```

```

DoCmd.PrintOut
Exit_btnPrint_Click:
Exit Sub
Err_btnPrint_Click:
MsgBox Err.Description
Resume Exit_btnPrint_Click
End Sub

```

```

Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
If Me.Dirty Then Me.Dirty = False
DoCmd.Close
Exit_btnTutup_Click:
Exit Sub
Err_btnTutup_Click:
MsgBox Err.Description
Resume Exit_btnTutup_Click
End Sub

```

21. Kode Program *Form* P37Pareto

```

Option Compare Database
Private Sub Command1_Click()
Dim rs As Recordset

```

```
Dim sql As String
Dim pp As Pareto
sql = "Select * From P37Pareto"
Set rs = CurrentDb.OpenRecordset(sql)
rs.OpenRecordset
Set pp = Pareto0.Object
pp.Refresh
Dim satu As Double
Dim dua As Double
Dim tiga As Double
Dim empat As Double
Dim lima As Double
Dim enam As Double
satu = rs("SumOfoktaf_aneh")
dua = rs("SumOfBiri")
tiga = rs("SumOfReed_tinggi")
empat = rs("SumOfReed_rendah")
lima = rs("SumOfBunyi_tersumbat")
enam = rs("SumOfTidak_bunyi")
pp.Y1AxisText = PShowCountAxis
pp.Y1AxisInterval = 10
pp.Y2AxisText = PShowPercentAxis
pp.Y2AxisInterval = 10
pp.Count(1) = satu
pp.Color(1) = vbMagenta
pp.Category(1) = "oktaf aneh"
pp.Count(2) = dua
pp.Color(2) = vbBlue
pp.Category(2) = "Biri"
pp.Count(3) = tiga
pp.Color(3) = vbGreen
pp.Category(3) = "Reed tinggi"
pp.Count(4) = empat
pp.Color(4) = vbRed
```

```
pp.Category(4) = "Reed rendah"  
pp.Count(5) = lima  
pp.Color(5) = vbBlack  
pp.Category(5) = "Bunyi tersumbat"  
pp.Count(6) = enam  
pp.Color(6) = vbMagenta  
pp.Category(6) = "Tidak bunyi"  
pp.GraphContent = PCount  
pp.Refresh  
rs.Close  
Set rs = Nothing  
End Sub
```

```
Private Sub btnMainMenu_Click()  
On Error GoTo Err_btnMainMenu_Click  
Dim stDocName As String  
Dim stLinkCriteria As String  
stDocName = "Switchboard"  
DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_btnMainMenu_Click:  
Exit Sub  
Err_btnMainMenu_Click:  
MsgBox Err.Description  
Resume Exit_btnMainMenu_Click  
End Sub
```

```
Private Sub btnKeluar_Click()  
On Error GoTo Err_btnKeluar_Click  
DoCmd.Close  
Exit_btnKeluar_Click:  
Exit Sub  
Err_btnKeluar_Click:  
MsgBox Err.Description  
Resume Exit_btnKeluar_Click
```

End Sub

22. Kode Program *Form* Pegawai

Option Compare Database

Private Sub btnMainMenu_Click()

On Error GoTo Err_btnMainMenu_Click

Dim stDocName As String

Dim stLinkCriteria As String

stDocName = "Switchboard"

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_btnMainMenu_Click:

Exit Sub

Err_btnMainMenu_Click:

MsgBox Err.Description

Resume Exit_btnMainMenu_Click

End Sub

Private Sub btnbaru_Click()

On Error GoTo Err_btnbaru_Click

DoCmd.GoToRecord , , acNewRec

Exit_btnbaru_Click:

Exit Sub

Err_btnbaru_Click:

MsgBox Err.Description

Resume Exit_btnbaru_Click

End Sub

Private Sub btnSimpan_Click()

On Error GoTo Err_btnSimpan_Click

DoCmd.RunCommand acCmdSaveRecord

Exit_btnSimpan_Click:

Exit Sub

```
Err_btnSimpan_Click:
    MsgBox Err.Description
    Resume Exit_btnSimpan_Click
End Sub

Private Sub btnHapus_Click()
On Error GoTo Err_btnHapus_Click
    DoCmd.RunCommand acCmdSelectRecord
    DoCmd.RunCommand acCmdDeleteRecord
Exit_btnHapus_Click:
    Exit Sub
Err_btnHapus_Click:
    MsgBox Err.Description
    Resume Exit_btnHapus_Click
End Sub

Private Sub btnPrev_Click()
On Error GoTo Err_btnPrev_Click
    DoCmd.GoToRecord , , acPrevious
Exit_btnPrev_Click:
    Exit Sub
Err_btnPrev_Click:
    MsgBox Err.Description
    Resume Exit_btnPrev_Click
End Sub

Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
    DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
    Exit Sub
Err_btnNext_Click:
    MsgBox Err.Description
    Resume Exit_btnNext_Click
End Sub
```



```

Private Sub btnTutup_Click()
On Error GoTo Err_btnTutup_Click
If Me.Dirty Then Me.Dirty = False
DoCmd.Close
Exit_btnTutup_Click:
Exit Sub
Err_btnTutup_Click:
MsgBox Err.Description
Resume Exit_btnTutup_Click
End Sub

```

```

Private Sub Combo18_AfterUpdate()
Dim rs As Object
Set rs = Me.Recordset.Clone
rs.FindFirst "[NIK] = " & Me![Combo18] & ""
If Not rs.EOF Then Me.Bookmark = rs.Bookmark
End Sub

```

23. Kode Program *Form POP_UP_ES*

```

Option Compare Database
Private Sub Command1_Click()
On Error GoTo Err_Command1_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "frm_input"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command1_Click:
Exit Sub
Err_Command1_Click:
MsgBox Err.Description
Resume Exit_Command1_Click
End Sub

```

24. Kode Program *Form* Proses Kritis

```
Option Compare Database
Private Sub btnbaru_Click()
On Error GoTo Err_btnbaru_Click
DoCmd.GoToRecord , , acNewRec
Exit_btnbaru_Click:
Exit Sub
Err_btnbaru_Click:
MsgBox Err.Description
Resume Exit_btnbaru_Click
End Sub

Private Sub btnSimpan_Click()
On Error GoTo Err_btnSimpan_Click
DoCmd.RunCommand acCmdSaveRecord
Exit_btnSimpan_Click:
Exit Sub
Err_btnSimpan_Click:
MsgBox Err.Description
Resume Exit_btnSimpan_Click
End Sub

Private Sub btnHapus_Click()
On Error GoTo Err_btnHapus_Click
DoCmd.RunCommand acCmdSelectRecord
DoCmd.RunCommand acCmdDeleteRecord
Exit_btnHapus_Click:
Exit Sub
Err_btnHapus_Click:
MsgBox Err.Description
Resume Exit_btnHapus_Click
End Sub
```

```
Private Sub btnPrev_Click()  
On Error GoTo Err_btnPrev_Click  
DoCmd.GoToRecord , , acPrevious  
Exit_btnPrev_Click:  
Exit Sub  
Err_btnPrev_Click:  
MsgBox Err.Description  
Resume Exit_btnPrev_Click  
End Sub
```

```
Private Sub btnNext_Click()  
On Error GoTo Err_btnNext_Click  
DoCmd.GoToRecord , , acNext  
Exit_btnNext_Click:  
Exit Sub  
Err_btnNext_Click:  
MsgBox Err.Description  
Resume Exit_btnNext_Click  
End Sub
```

25. Kode Program *Form* Solusi

```
Option Compare Database  
Private Sub btnbaru_Click()  
On Error GoTo Err_btnbaru_Click  
DoCmd.GoToRecord , , acNewRec  
Exit_btnbaru_Click:  
Exit Sub  
Err_btnbaru_Click:  
MsgBox Err.Description  
Resume Exit_btnbaru_Click  
End Sub
```

```
Private Sub btnSimpan_Click()  
On Error GoTo Err_btnSimpan_Click  
    DoCmd.RunCommand acCmdSaveRecord  
Exit_btnSimpan_Click:  
    Exit Sub  
Err_btnSimpan_Click:  
    MsgBox Err.Description  
    Resume Exit_btnSimpan_Click  
End Sub
```

```
Private Sub btnHapus_Click()  
On Error GoTo Err_btnHapus_Click  
    DoCmd.RunCommand acCmdSelectRecord  
    DoCmd.RunCommand acCmdDeleteRecord  
Exit_btnHapus_Click:  
    Exit Sub  
Err_btnHapus_Click:  
    MsgBox Err.Description  
    Resume Exit_btnHapus_Click  
End Sub
```

```
Private Sub btnTutup_Click()  
On Error GoTo Err_btnTutup_Click  
    If Me.Dirty Then Me.Dirty = False  
    DoCmd.Close  
Exit_btnTutup_Click:  
    Exit Sub  
Err_btnTutup_Click:  
    MsgBox Err.Description  
    Resume Exit_btnTutup_Click  
End Sub
```

```
Private Sub btnPrev_Click()  
On Error GoTo Err_btnPrev_Click
```

```

DoCmd.GoToRecord , , acPrevious
Exit_btnPrev_Click:
Exit Sub
Err_btnPrev_Click:
MsgBox Err.Description
Resume Exit_btnPrev_Click
End Sub

```

```

Private Sub btnNext_Click()
On Error GoTo Err_btnNext_Click
DoCmd.GoToRecord , , acNext
Exit_btnNext_Click:
Exit Sub
Err_btnNext_Click:
MsgBox Err.Description
Resume Exit_btnNext_Click
End Sub

```

```

Private Sub Combo4_AfterUpdate()
Dim rs As Object
Set rs = Me.Recordset.Clone
rs.FindFirst "[ID_solusi] = " & Me![Combo4] & ""
If Not rs.EOF Then Me.Bookmark = rs.Bookmark
End Sub

```

26. Kode Program *Form* Main Menu (*form* Switchboard)

```

Option Compare Database
Private Sub Form_Load()
If Admin.Value = "Admin" Or Admin.Value = "Guest" Then
Exit Sub
Else
x = InputBox("Masukkan password", "Administrator")
If x = "admin" Then

```

```
Admin.Value = "Admin"  
Exit Sub  
Else  
Option6.Enabled = False  
Admin.Value = "Guest"  
End If  
End If  
End Sub
```

```
Private Sub Form_Open(Cancel As Integer)  
' Minimize the database window and initialize the form.  
' Move to the switchboard page that is marked as the default.  
Me.Filter = "[ItemNumber] = 0 AND [Argument] = 'Default' "  
Me.FilterOn = True  
End Sub
```

```
Private Sub Form_Current()  
' Update the caption and fill in the list of options.  
Me.Caption = Nz(Me![ItemText], "")  
FillOptions  
End Sub
```

```
Private Sub FillOptions()  
' Fill in the options for this switchboard page.  
' The number of buttons on the form.  
Const conNumButtons = 8  
Dim con As Object  
Dim rs As Object  
Dim stSql As String  
Dim intOption As Integer  
' Set the focus to the first button on the form,  
' and then hide all of the buttons on the form  
' but the first. You can't hide the field with the focus.  
Me![Option1].SetFocus
```

```

For intOption = 2 To conNumButtons
    Me("Option" & intOption).Visible = False
    Me("OptionLabel" & intOption).Visible = False
Next intOption
' Open the table of Switchboard Items, and find
' the first item for this Switchboard Page.
Set con = Application.CurrentProject.Connection
stSql = "SELECT * FROM [Switchboard Items]"
stSql = stSql & " WHERE [ItemNumber] > 0 AND [SwitchboardID]=" &
Me![SwitchboardID]
stSql = stSql & " ORDER BY [ItemNumber];"
Set rs = CreateObject("ADODB.Recordset")
rs.Open stSql, con, 1 ' 1 = adOpenKeyset
' If there are no options for this Switchboard Page,
' display a message. Otherwise, fill the page with the items.
If (rs.EOF) Then
    Me![OptionLabel1].Caption = "There are no items for this switchboard page"
Else
    While (Not (rs.EOF))
        Me("Option" & rs![ItemNumber]).Visible = True
        Me("OptionLabel" & rs![ItemNumber]).Visible = True
        Me("OptionLabel" & rs![ItemNumber]).Caption = rs![ItemText]
        rs.MoveNext
    Wend
End If
' Close the recordset and the database.
rs.Close
Set rs = Nothing
Set con = Nothing
End Sub

Private Function HandleButtonClick(intBtn As Integer)
' This function is called when a button is clicked.
' intBtn indicates which button was clicked.

```

```
' Constants for the commands that can be executed.
Const conCmdGotoSwitchboard = 1
Const conCmdOpenFormAdd = 2
Const conCmdOpenFormBrowse = 3
Const conCmdOpenReport = 4
Const conCmdCustomizeSwitchboard = 5
Const conCmdExitApplication = 6
Const conCmdRunMacro = 7
Const conCmdRunCode = 8
Const conCmdOpenPage = 9
' An error that is special cased.
Const conErrDoCmdCancelled = 2501
Dim con As Object
Dim rs As Object
Dim stSql As String
On Error GoTo HandleButtonClick_Err
' Find the item in the Switchboard Items table
' that corresponds to the button that was clicked.
Set con = Application.CurrentProject.Connection
Set rs = CreateObject("ADODB.Recordset")
stSql = "SELECT * FROM [Switchboard Items] "
stSql = stSql & "WHERE [SwitchboardID]=" & Me![SwitchboardID] & " AND
[ItemNumber]=" & intBtn
rs.Open stSql, con, 1 ' 1 = adOpenKeyset
' If no item matches, report the error and exit the function.
If (rs.EOF) Then
    MsgBox "There was an error reading the Switchboard Items table."
    rs.Close
    Set rs = Nothing
    Set con = Nothing
    Exit Function
End If
Select Case rs![Command]
    ' Go to another switchboard.
```


Case conCmdGotoSwitchboard

```
Me.Filter = "[ItemNumber] = 0 AND [SwitchboardID]=" & rs![Argument]
```

' Open a form in Add mode.

Case conCmdOpenFormAdd

```
DoCmd.OpenForm rs![Argument], , , , acAdd
```

' Open a form.

Case conCmdOpenFormBrowse

```
DoCmd.OpenForm rs![Argument]
```

' Open a report.

Case conCmdOpenReport

```
DoCmd.OpenReport rs![Argument], acPreview
```

' Customize the Switchboard.

Case conCmdCustomizeSwitchboard

```
' Handle the case where the Switchboard Manager  
' is not installed (e.g. Minimal Install).
```

```
On Error Resume Next
```

```
Application.Run "ACWZMAIN.sbm_Entry"
```

```
If (Err <> 0) Then MsgBox "Command not available."
```

```
On Error GoTo 0
```

```
' Update the form.
```

```
Me.Filter = "[ItemNumber] = 0 AND [Argument] = 'Default' "
```

```
Me.Caption = Nz(Me![ItemText], "")
```

```
FillOptions
```

' Exit the application.

Case conCmdExitApplication

```
CloseCurrentDatabase
```

' Run a macro.

Case conCmdRunMacro

```
DoCmd.RunMacro rs![Argument]
```

' Run code.

Case conCmdRunCode

```
Application.Run rs![Argument]
```

' Open a Data Access Page

Case conCmdOpenPage

```
DoCmd.OpenDataAccessPage rs![Argument]
' Any other command is unrecognized.
Case Else
    MsgBox "Unknown option."
End Select
' Close the recordset and the database.
rs.Close
HandleButtonClick_Exit:
On Error Resume Next
Set rs = Nothing
Set con = Nothing
Exit Function
HandleButtonClick_Err:
' If the action was cancelled by the user for
' some reason, don't display an error message.
' Instead, resume on the next line.
If (Err = conErrDoCmdCancelled) Then
    Resume Next
Else
    MsgBox "There was an error executing the command.", vbCritical
    Resume HandleButtonClick_Exit
End If
End Function
```