BAB II

LANDASAN TEORI

2.1 Aplikasi

Aplikasi adalah penerapan, menyimpan sesuatu data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, pekerjaan itu sendiri (Jogiyanto, 2005).

2.2 Penjadwalan

Penjadwalan adalah pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin. Dengan demikian masalah *sequencing* senantiasa melibatkan pengerjaan sejumlah komponen yang sering disebut dengan istilah '*job*'. Job masih merupakan komposisi dari sejumlah elemen-elemen dasar yang disebut aktivitas atau operasi. Tiap aktivitas atau perasi ini membutuhkan alokasi sumber daya tertentu selama periode waktu tertentu yang sering disebut dengan waktu proses (Ginting, 2007).

Penjadwalan dapat diartikan sebagai proses pengalokasian sumber daya yang terbatas untuk memilih tugas dalam jangka waktu tertentu. Definisi umum ini memiliki dua arti. Pertama, penjadwalan merupakan fungsi dari pembuat keputusan yang merupakan proses untuk menetapkan suatu jadwal sehingga dapat diterapkan pada proses pembuatan keputusan yang lain. Kedua, Penjadwalan sebagai suatu teori yang terdiri dari pengumpulan dasar, model, teknik dan konklusi logis yang memberikan gambaran dalam fungsi penjadwalan.

2.3 Penjadwalan Produksi

Penjadwalan produksi secara umum didefinisikan sebagai suatu proses dalam perencanaan dan pengendalian produksi dan pengalokasian sumber daya pada suatu waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada (Tanuwijaya, 2012).

Secara umum, penjadwalan produksi dapat dibedakan menjadi dua jenis yaitu penjadwalan per job dan penjadwalan per *batch*. Berdasarkan tahapan proses produksinya penjadwalan per job dibedakan menjadi dua yaitu *single stage* dan *multiple stage*. Berdasarkan jumlah mesin yang digunakan dalam proses produksi, penjadwalan *single stage* dibedakan menjadi dua jenis yaitu *single machine* dan *paralel machine* (Pinedo, 2002).

2.3.1 Klasifikasi Penjadwalan Produksi

Pada dasarnya terdapat beberapa klasifikasi yang dapat digunakan untuk menetapkan prioritas dalam operasi manufakturing, antara lain. (Gaspersz, 2004).

1. *Critical Ratios*, dihitung melalui pembagian waktu yang tersisa (banyaknya jam atau hari kerja di antara sekarang dan *due date*) dengan kerja (*manufacturing time*) yang tersisa (*total setup, run, wait, move and queue times*).
2. *Shortest Processing Time* (SPT). pesanan-pesanan dengan jumlah *setup and run time* yang dibutuhkan pada *current work center* terkecil adalah yang diprioritaskan untuk dikerjakan lebih dulu. Pesanan-pesanan yang memiliki waktu pemrosesan terpendek memiliki proritas lebih tinggi untuk dikerjakan terlebih dahulu pada *current work center*. Aturan ini dapat menunda pekerjaan-pekerjaan yang mempunyai waktu proses panjang,

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

sehingga direkomendasikan untuk digunakan secara sementara saja, dan bukan merupakan aturan yang tetap dalam menentukan prioritas.

3. *First-Come, First-Server* (FCFS). Tugas yang pertama datang ke pusat kerja diproses terlebih dahulu. Metode ini hanya digunakan apabila waktu kerja yang tersisa untuk *competing orders* relatif sama. FCFS akan cocok untuk *flow processes* karena memiliki *work remaining times* yang serupa.
4. *Earliest Due Date* (EDD). Tugas-tugas yang mempunyai *earliest due date* yang dipilih pertama.

   Longest Processing Time (LPT). Tugas-tugas yang mempunyai waktu proses terpanjang yang dipilih terlebih dahulu.

2.3.2 Kriteria Pemilihan Metode Penjadwalan

Terdapat kriteria-kriteria yang dapat digunakan sebagai dasar pemilihan metode penjadwalan yang sesuai yakni (Nasution, 2003).

1. *Mean flow time* : Rata-rata waktu tinggal pekerjaan dalam sistem. Biasanya menggunakan metode LPT lalu dilanjutkan dengan SPT.
2. *Makespan* : Waktu penyelesaian semua pekerjaan. Sama seperti *mean flow time*, pemecahannya menggunakan aturan slack.
3. *Tardiness* : Keterlambatan. Pemecahannya menggunakan aturan slack.
4. *Mean tardiness* : Rata-rata waktu keterlambatan. Untuk mengurangi *mean tardiness* menggunakan metode SPT, EDD dan slack lalu diilanjutkan dengan algoritma Wilkerson – Irwin.
5. *Maximum tardiness* : Keterlambatan maksimum. Untuk meminimalkan *maximum tardiness*, menggunakan metode EDD.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

6. *Number of tardy job* : Jumlah pekerjaan yang terlambat. Untuk mengurangi *number of tardy job*, menggunakan metode EDD lalu dilanjutkan dengan algoritma Hodgson.

2.3.3 Istilah – Istilah dalam Penjadwalan Produksi

Beberapa istilah umum yang digunakan dalam penjadwalan produksi yaitu (Nasution, 2003).

1. *Processing Time* (waktu proses), merupakan perkiraan waktu penyelesaian satu pekerjaan. Perkiraan meliputi perkiraan waktu setup mesin. Simbol untuk waktu proses pekerjaan i adalah Ti.
2. *Due date* (batas waktu), merupakan waktu maksimal yang dapat diterima untuk menyelesaikan pekerjaan tersebut. Kelebihan waktu dari waktu yang telah ditetapkan merupakan suatu keterlambatan. Batas waktu ini disimbolkan dengan d_i.
3. *Lateness* (keterlambatan), merupakan penyimpangan antara waktu penyelesaian pekerjaan dengan waktu yang ditentukan. Suatu pekerjaan mempunyai keterlambatan positif jika diselesaikan setelah batas waktu dan bernilai negatif jika diselesaikan sebelum batas waktu. Simbol keterlambatan ini adalah L.

$$L_i = C_i - d_i \quad \text{..... (2.1)}$$

(Sumber : Nasution, 2003)

4. *Tardiness* (ukuran keterlambatan), merupakan ukuran untuk keterlambatan positif. Jika suatu pekerjaan diselesaikan lebih cepat dari batas waktu yang ditetapkan, maka mempunyai nilai keterlambatan negatif tetapi ukuran

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

keterlambatan positif. Ukuran ini disimbolkan dengan T_i dimana T_i adalah maksimum dari (0, L_i).

5. *Slack* (kelonggaran), merupakan ukuran yang digunakan untuk melihat selisih waktu antara waktu proses dengan batas waktu yang telah ditetapkan. Slack dinotasikan Sl_i dan dihitung dengan persamaan.

$$Sl_i = d_i - t_i \quad \text{..... (2.2)}$$

(Sumber : Nasution, 2003)

6. *Completion time* (waktu penyelesaian), merupakan rentang waktu antara saat pekerjaan dimulai sampai pekerjaan itu selesai. Waktu penyelesaian ini disimbolkan dengan C_i.

7. *Flow Time* (waktu alir), merupakan rentang waktu antara saat pekerjaan dapat dimulai (tersedia) dan saat pekerjaan selesai. Waktu alir sama dengan waktu proses ditambah dengan waktu tunggu sebelum pekerjaan diproses.

2.4 Metode *Earliest Due Date*

Metode *Earliest Due Date* merupakan pengurutan pekerjaan-pekerjaan berdasarkan tanggal jatuh tempo (*due date*) yang tercepat. Pekerjaan dengan saat jatuh tempo paling awal harus dijadwalkan terlebih dahulu daripada pekerjaan dengan saat jatuh tempo belakangan. Metode ini dapat digunakan untuk penjadwalan pada satu mesin (*single machine*) maupun untuk penjadwalan pada beberapa mesin (*paralel machine*). Metode penjadwalan yang menghasilkan *maximum tardiness* yang paling kecil minimum adalah metode *Earliest Due Date* (Kusuma, 2009).

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

Dalam prosedur jatuh tempo, pekerjaan diurutkan berdasarkan jatuh tempo terdekat atau berdasarkan tugas-tugas yang mempunyai tanggal yang dibutuhkan paling cepat. Prosedur jatuh tempo meminimalkan keterlambatan (*tardiness*) maksimum.

Langkah – langkah penggunaan metode EDD adalah sebagai berikut.

Langkah 1 : Urutkan pekerjaan berdasarkan tanggal jatuh tempo terdekat.

Langkah 2 : Ambil pekerjaan satu persatu dari urutan berdasarkan tanggal jatuh tempo itu, kemudian jadwalkan pada mesin dengan beban yang paling minimum.

Parameter-parameter yang diperlukan dalam penjadwalan dengan metode *Earliest Due Date* (EDD) yaitu.

a. Waktu pemrosesan

b. Batas waktu tiap pekerjaan

Setelah dibuat urutan berdasarkan waktu pemrosesan, maka EDD akan menghasilkan perhitungan ukuran efektivitas yang dapat dilihat pada Gambar 1.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

$$\text{Rata-rata waktu penyelesaian} = \frac{\text{jumlah total waktu aliran}}{\text{jumlah pekerjaan}} = \ldots. \text{ hari} \quad \ldots\ldots\ldots (2.3)$$

$$\text{Utilisasi} = \frac{\text{total waktu pekerjaan}}{\text{jumlah total waktu aliran}} = \ldots. \% \quad \ldots\ldots\ldots (2.4)$$

$$\text{Rata - rata jumlah pekerjaan} = \frac{\text{jumlah total waktu aliran}}{\text{total waktu pekerjaan}} = \ldots \text{ pekerjaan} \quad \ldots\ldots\ldots (2.5)$$

$$\text{Rata – rata keterlambatan pekerjaan} = \frac{\text{total hari terlambat}}{\text{jumlah pekerjaan}} = \ldots. \text{ hari} \quad \ldots\ldots\ldots (2.6)$$

Gambar 2.1 Perhitungan Metode *Earliest Due Date* (Kusuma, 2009).

Studi Kasus Penerapan Metode EDD

Berikut merupakan contoh perhitungan produksi pupuk phospat dengan kadar 18%, rata-rata produksi pupuk phospat yang dihasilkan sekitar 250 ton per minggu dan per hari dapat menghasilkan pupuk sebesar 45 ton. Data perhitungan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Data Perhitungan

Urutan Pekerjaan (Nama Customer)	Jumlah Pesan (Ton)	Waktu Pemrosesan (Hari)	Batas Waktu Pekerjaan (Hari)
PT. Rekayasa Industri	60	2	1
PT. Mega Eltra	120	3	4
PT. Pupuk Sriwidjaya	70	2	2

Penyelesaian perhitungan dengan metode EDD dapat dilihat pada Tabel 2.2 :

Tabel 2.2 Perhitungan Metode EDD

Urutan Pekerjaan (Nama Customer)	Jumlah Pesan (Ton)	Waktu Pemrosesan (Hari)	Aliran Waktu (Hari)	Batas Waktu Pekerjaan (Hari)	Keterlambatan (Hari)
PT. Rekayasa Industri	60	2	2	1	1
PT Pupuk Sriwidjaya	70	2	4	2	2
PT Mega Eltra	120	3	7	4	3
Jumlah	250	7	13		6

Metode EDD menghasilkan ukuran efektivitas seperti pada persamaan 2.3, 2.4, 2.5, 2.6.

a. Rata-rata waktu penyelesaian = jumlah total waktu aliran / jumlah pekerjaan

= 13 hari / 5 = 2,6 hari

b. Utilisasi = total waktu pekerjaan / jumlah total waktu aliran

= 7 / 13 = 0,53 %

c. Rata - rata jumlah pekerjaan = jumlah total waktu aliran / total waktu Pekerjaan

= 13 hari / 7 hari = 1,85 pekerjaan

d. Rata – rata keterlambatan pekerjaan = total hari terlambat / jumlah Pekerjaan

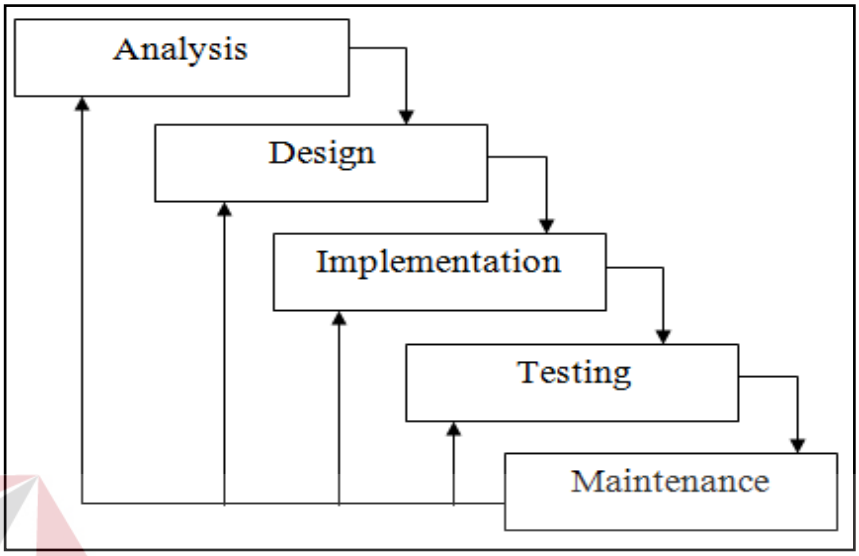
= 6 / 3 = 2 hari

2.5 *System Development Life Cycle*

System development life cycle (SDLC) adalah keseluruhan proses dalam membangun sistem informasi melalui beberapa tahapan. Ada beberapa model SDLC namun yang paling sering digunakan dan paling populer adalah model *waterfall*. Model ini disebut *waterfall* karena dikerjakan langkah per langkah seperti air mengalir.Adapun model lain dari SDLC *yaitu fountain, spiral, rapid prototyping, incremental, build & fix, System development life cycle* dan *synchronize & stabilize*.

Model *waterfall* adalah proses pengembangan perangkat lunak sekuensial dimana kemajuan dianggap mengalir semakin ke bawah melalui daftar tahan yang harus dijalankan agar berhasil membangun sebuah perangkat lunak komputer. Model *waterfall* mendefinisikan beberapa fase yang berurutan yang harus diselesaikan satu demi satu dan pindah ke tahap berikutnya, hanya ketika fase

sebelumnya yang benar-benar dilakukan. Gambar 2.2 merupakan tahapan dalam membangun sistem informasi menggunakan model *waterfall* (Bassil, 2011).



Gambar 2.2 Tahapan Membangun Sistem Informasi (Bassil, 2011).

1. *Analysis*

Fase analisis adalah deskripsi lengkap dari perilaku perangkat lunak yang akan dikembangkan. Sistem dan bisnis analisis untuk menentukan persyaratan fungsional dan non-fungsional. Persyaratan fungsional didefinisikan dengan cara menggunakan kasus yang menggambarkan interaksi pengguna dengan perangkat lunak seperti tujuan, ruang lingkup, perspektif, fungsi, atribut perangkat lunak, karakteristik pengguna, spesifikasi fungsi, dan persyaratan antarmuka. Sebaliknya, persyaratan non-fungsional mengacu pada berbagai kriteria, kendala, keterbatasan dan persyaratan yang diberlakukan pada desain dan pengoperasian pernagkat lunak seperti keandalan, skalabilitas, ketersediaan, kinerja, dan standar kualitas.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

2. *Design*

Fase desain adalah proses perencanaan dan pemecahan solusi masalah perangkat lunak untuk menentukan rencana untuk solusi yang meliputi desain algoritma, desain arsitektur perangkat lunak, skema database konseptual, desain diagram logis, desain konsep, desain antarmuka pengguna grafis, dan struktur data.

3. *Implementation*

Fase implementasi mengacu pada realisasi kebutuhan bisnis dan spesifikasi desain ke dalam program, *database*, atau komponen perangkat lunak melalui pemrograman. Fase ini adalah dimana kode nya ditulis dan disusun menjadi sebuah aplikasi operasional, dan dimana *database* dan file teks yang dibuat.

4. *Testing*

Fase *testing* dikenal sebagai verifikasi dan validasi yang merupakan proses untuk memeriksa bahwa solusi perangkat lunak memenuhi persyaratan asli dan spesifikasi bahwa itu dapat menyelesaikan tujuan yang telah ditetapkan. Verifikasi adalah proses evaluasi *software* untuk menentukan apakah produk dari tahap pengembangan yang diberikan dapat memenuhi kondisi yang dikenakan pada awal fase. Validasi adalah proses evaluasi *software* pada akhir proses pengembangan untuk menentukan apakah dapat menentukan persyaratan yang ditentukan. Tahap pengujian adalah outlet untuk melakukan debugging dimana bug dan ganggunan sistem ditemukan, dikoreksi, dan disempurnakan kembali.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

5. *Maintenance*

Fase pemeliharaan adalah proses memodifikasi solusi perangkat lunak setelah memperbaiki *output*, memperbaiki kesalahan, dan meningkatkan kinerja serta kualitas. Kegiatan pemeliharaan tambahan dapat dilakukan dalam fase ini termasuk meningkatkan kehandalan *software*.

2.6 *Microsoft Visual Basic.Net*

Visual Basic adalah bahasa pemrograman tingkat tinggi yang sudah sangat terkenal, dimulai dengan BASIC yang terdapat pada computer "angkatan tua" seperti AT286. Pada saat itu bahasa BASIC merupakan bahasa yang sangat diandalkan dalam pembuatan beberapa aplikasi penting. BASIC digemari karena susunan programnya yang membebaskan kita untuk "melompat" dari satu baris ke baris yang lainnya. Versi BASIC lainnya adalah BASICA, Qbasic, Turbi Basic dan lain-lain. Bahasa BASIC banyak terdapat di masa penggunaan sistem operasi DOS (Wahana Komputer, 2009).

2.7 *SQL Server*

SQL Server adalah sebuah sistem arsitektur terbuka yang memungkinkan para pengembang program memperluas dan menambahkan fungsi-fungsi ke dalam *database* (Djuandi, 2002).

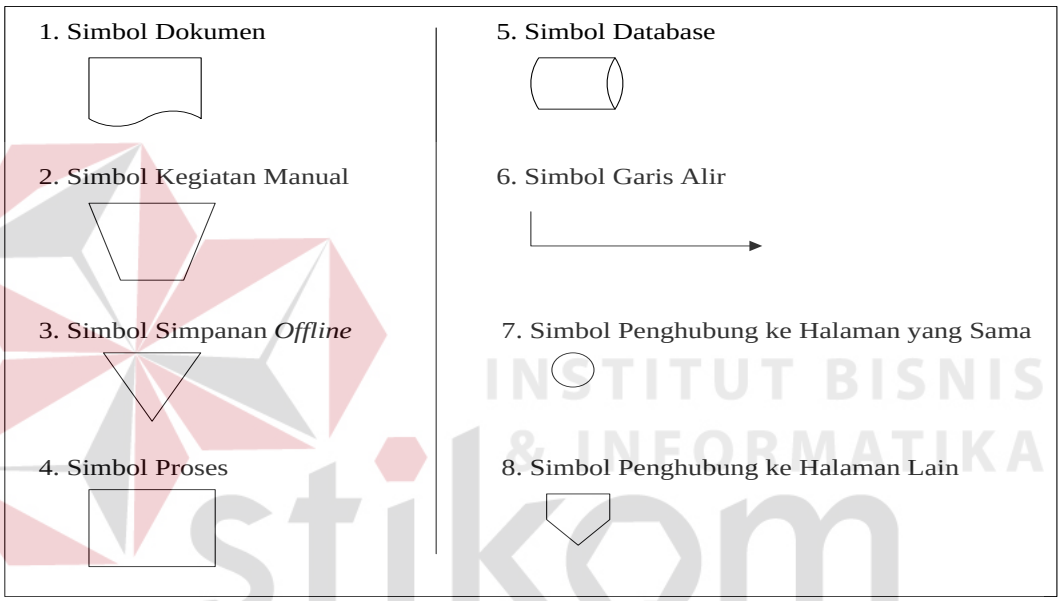
SQL Server adalah perangkat lunak *Relation Database Management System* (RDBMS) yang di desain untuk melakukan proses manipulasi database berukuran besar dengan berbagai fasilitas (Kuniyo, 2007).

Dari pengertian diatas, maka dapat disimpulkan SQL Server merupakan suatu *software* RDMS yang didesain untuk para pengembang program guna untuk

melakukan proses manipulasi, memperluas database dengan berbagai fitur yang terdapat di dalamnya.

2.8 *System Flow*

System flow menunjukan urutan- urutan dari prosedur yang ada di dalam sistem dan menunjukan apa yang dikerjakan sistem. Simbol- simbol yang digunakan dalam *System flow* ditunjukan pada Gambar 2.3.



Gambar 2.3 Simbol *System Flow* (Jogiyanto, 2005).

2.9 *Data Flow Diagram*

Data flow diagram (DFD) awalnya dikembangkan oleh Chris Ganeda Trish Sarson pada tahun 1979 yang termasuk dalam *Strucutred Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem.

Data flow diagram (DFD) dapat digunakan untuk merepresentaikan sebuah sistem atau perangkat lunak pada beberapa level ambstraksi. DFD dapat

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

dibagi menjadi beberapa level yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Simbol-simbol dasar dalam DFD meliputi.

1. *External Entity*

Sebuah elemen sistem atau sistem yang lain yang menghasilkan informasi bagi transformasi oleh perangkat lunak atau menerima informasi yang dihasilkan oleh perangkat lunak.

2. *Data Flow*

Data Flow atau aliran data disimbolkan dengan tanda panah. Data Flow menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses.

3. *Process*

Mempresentasikan sebuah proses atau transformasi yang diaplikasikan ke data untuk mengubahnya dengan berbagai macam cara.

4. *Data Store*

Simbol *data store* merupakan simbol yang digunakan untuk melambangkan proses penyimpanan data.

2.10 *Entity Relation Diagram*

Entity relationship diagram adalah suatu bentuk perencanaan database secara konsep fisik yang nantinya akan dipakai sebagai kerangka kerja dan pedoman dari struktur penyimpanan data. ERD digunakan untuk menggambarkan model hubungan data dalam sistem, dimana di dalamnya terdapat hubungan entitas beserta atribut relasinya dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan data (Jogiyanto, 2005).

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

Beberapa jenis model ERD, yaitu :

1. *Conceptual Data Model* (CDM)

Merupakan model *universal* dan dapat menggambarkan semua struktut logic database (DBMS), dan tidak bergantung dari software atau pertimbangan struktur data *storage*. Sebuah CDM dapat diubah langsung menjadi PDM.

2. *Physical Data Model* (PDM)

Merupakan model ERD yang mengacu pada pemilihan *software* DBMS yang spesifik. Hal ini seringkali berbeda secara signifikan dikarenakan oleh struktur tipe database yang bervariasi, dari model schema, tipe data penyimpanan dan sebagainya.

2.11 *Crystal Report*

Crystal Report adalah program yang dapat digunakan untuk membuat, menganalilis dan menterjemahkan informasi yang terkandung dalam database atau program ke dalam berbagai jenis laporan yang sangat flexibel. (Kuniyo, 2007).

Crystal Report merupakan program khusus untuk membuat laporan terpisah dari program microsoft Visual Basic 6.0, tetapi keduanya dapat dihubungkan (*linkage*). (Madcom, 2003).

Beberapa fungsi *tools* yang ada di *Crystal Report* :

a. *Report Header*, digunakan untuk informasi yang ditampilkan pada halaman pertama saja. Contohnya logo dan kop surat yang tertletak di posisi atas.

b. *Page Header*, digunakan untuk informasi yang ditampilkan pada setiap halaman. Contohnya nama kolom.

c. *Group Header*, area informasi yang terletak dibawah page header.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

d. *Detail*, area yang digunakan untuk menampilkan isi datanya.

e. *Report Footer*, digunakan untuk informasi yang ditampilkan pada halaman terakhir. Contohnya tanda tangan, nama penanggung jawab.

f. *Page Footer*, digunakan untuk menampilkan halaman.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA