

BAB II

LANDASAN TEORI

2.1 Pengertian Penilaian

Penilaian adalah proses memberikan atau menentukan nilai kepada objek tertentu berdasarkan suatu criteria tertentu [1]. Menurut Mehrens dan Lehmans [2] penilaian adalah suatu proses merencanakan, dan menyediakan informasi yang sangat diperlukan untuk membuat alternative-alternatif keputusan.

2.2 Perilaku kerja

Perilaku kerja pegawai tercermin melalui uraian pekerjaan yang dimiliki oleh tiap-tiap pegawai. Pada Peraturan Pemerintah Republik Indonesia Nomor 46 Tahun 2011 dijelaskan definisi perilaku kerja adalah setiap tingkah laku, sikap atau tindakan yang dilakukan oleh pegawai atau tidak melakukan sesuatu yang seharusnya dilakukan sesuai dengan ketentuan peraturan perundang-undangan. Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa perilaku kerja akan menjadi tidak terarah apabila pegawai tersebut tidak memiliki pedoman yang jelas dalam menjalankan pekerjaannya dalam hal ini adalah deskripsi pekerjaan yang jelas dan lengkap. [3]

Perilaku kerja dapat dinilai berdasarkan beberapa aspek, antara lain :

1. Orientasi pelayanan

Orientasi pelayanan adalah sikap dan perilaku kerja Pegawai Negeri Sipil dalam memberikan pelayanan terbaik kepada yang dilayani antara lain meliputi masyarakat, atasan, rekan sekerja, unit kerja terkait, dan/atau instansi lain.

2. Integritas

Integritas adalah kemampuan untuk bertindak sesuai dengan nilai, norma dan etika dalam organisasi.

3. Komitmen

Komitmen adalah kemauan dan kemampuan untuk menyelaraskan sikap dan tindakan Pegawai Negeri Sipil untuk mewujudkan tujuan organisasi dengan mengutamakan kepentingan dinas daripada kepentingan diri sendiri, seseorang, dan/atau golongan.

4. Disiplin

Disiplin adalah kesanggupan Pegawai Negeri Sipil untuk menaati kewajiban dan menghindari larangan yang ditentukan dalam peraturan perundang-undangan dan/atau peraturan kedinasan yang apabila tidak ditaati atau dilanggar dijatuhi hukuman disiplin.

5. Kerjasama

Kerjasama adalah kemauan dan kemampuan Pegawai Negeri Sipil untuk bekerja sama dengan rekan sekerja, atasan, bawahan dalam unit kerjanya serta instansi lain dalam menyelesaikan suatu tugas dan tanggung jawab yang ditentukan, sehingga mencapai daya guna dan hasil guna yang sebesar-besarnya.

6. Kepemimpinan

Kepemimpinan adalah kemampuan dan kemauan Pegawai Negeri Sipil untuk memotivasi dan mempengaruhi bawahan atau orang lain yang berkaitan dengan bidang tugasnya demi tercapainya tujuan organisasi.

2.3 Penilaian Perilaku Kerja

Pengertian penilaian perilaku kerja adalah cara untuk melihat dan mengevaluasi tindakan yang ditunjukkan orang-orang yang bekerja di lingkungan kerja serta mengetahui sejauh mana tindakan tersebut dapat berperan/berpengaruh ditempat kerja.[4]

Kerberhasilan di berbagai wilayah kehidupan ternyata ditentukan oleh perilaku manusia, terutama perilaku kerja. Sebagian orang menyebut perilaku kerja ini sebagai motivasi, kebiasaan (habit) dan budaya kerja. Oleh karena itu penilaian diperlukan agar para karyawan dapat diupayakan untuk membentuk perilaku kerja yang konsisten dan positif. Perilaku kerja prestatif dapat dilihat dalam sikap sebagai berikut :

1. Kerja Ikhlas

Kerja ikhlas bukan berarti kerja tanpa mengharapkan gaji/honor. Kerja ikhlas dalam hal ini dapat diartikan kerja yang dilakukan tanpa keluh kesah. Segala jerihpayah bahkan rasa lelah tidak dirasakan suatu beban yang berat.

2. Kerja Mawas Diri dari Rasa Emosional

Kerja mawas diri dapat diartikan tidak tergesa-gesa dalam mengambil suatu tindakan, tidak mudah terpancing oleh suasana dalam menerima suatu kritikan maupun pujian. Sebelum bertindak dipikirkan dengan matang keputusan apa yang akan diambil. Oleh karena itu sikap hati-hati perlu diterapkan agar tidak mudah terjebak pada kesalahan yang sama.

3. Kerja Cerdas

Cerdas, sempurna perkembangan akal budinya (untuk berpikir, mengerti) dan tajam dalam berpikir. Bekerja tidak hanya mengandalkan otot saja tetapi juga mengandalkan otak artinya untuk mencapai sukses tidak hanya dibutuhkan kerja keras saja akan tetapi juga kecerdasan untuk melakukan inovasi baru yang dapat diterima oleh masyarakat. Kerja cerdas adalah bekerja dengan menggunakan pikiran yang tajam, cepat, tepat dalam menerima, menanggapi, menentukan sikap dan berbuat.

4. Kerja Keras

Kerja keras berarti bekerja dengan menggunakan sumber daya secara optimal, misalnya tenaga, pikiran, dan perasaan dalam menggunakan waktu, bahan, dan alat. Kerja keras dalam bekerja mempunyai sifat mabuk kerja untuk dapat mencapai sasaran yang ingin dicapai, dapat memanfaatkan

waktu yang optimal sehinggakadang-kadang tidak mengenal waktu, jarak, dan kesulitan yang dihadapi, sangat bersemangat untuk meraih keinginannya.

5. Kerja Tuntas

Kerja tuntas artinya kerja yang tidak setengah-setengah dan mampu mengorganisasikan bagian usaha secara terpadu dari awal sampai akhir untuk dapat menghasilkan usahanya secara maksimal.

2.4 Pengertian Sistem

Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai tujuan[5]. Didalam suatu sistem diperlukan adanya keterkaitan yang terdiri dari elemen-elemen yaitu:

1. Tujuan

Merupakan tujuan dari sistem tersebut dimana komputer digunakan untuk mengurangi dan membantu tugas-tugas yang dilakukan oleh manusia dalam pengolahan data.

2. Kontrol

Merupakan pengawasan dari pelaksanaan pencapaian tujuan sistem pengawas dapat berupa:

- a. Kontrol pemasukan data (*input*)
- b. Kontrol pengeluaran data (*output*)
- c. Kontrol pengoperasian (*processing*)

3. *Input*

Merupakan kajian dari sistem yang bertugas menerima data, dimana data yang masuk dapat berupa asal masukan dan jenis masukan.

4. Transformasi

Sistem komputer yang bertugas memproses data masukan (*input*) menjadi keluaran (*output*) sesuai dengan keinginan atau tujuan.

5. *Output*

Sistem komputer yang bertugas menghasilkan keluaran, tugasnya antara lain menghasilkan laporan dan grafik.

6. Umpan balik

Umpan balik bertujuan untuk melihat kembali apakah sistem telah berjalan sesuai dengan tujuan. Umpan balik yang dilakukan dapat berupa perbaikan dan pemeliharaan.

2.5 Pengertian Informasi

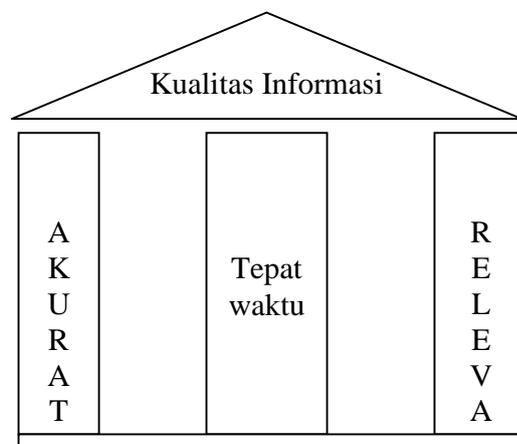
Terdapat beberapa definisi tentang pengertian informasi antara lain:

- a. Data yang di olah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima.
- b. Sesuatu yang nyata atau setengah nyata yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan atau kejadian.

Berdasarkan definisi diatas maka dapat disimpulkan informasi merupakan data yang telah diproses atau diolah yang memiliki arti penting bagi si penerima dan dapat mengurangi derajat ketidakpastian tentang suatu keadaan atau suatu kejadian.

2.5.1 Kualitas Informasi

John Burch dan *Gari Grudnitski* menggambarkan kualitas dari informasi dengan bentuk bangunan yang ditunjang oleh tiga buah pilar.



Gambar 2.1 : *Pilar kualitas informasi* [6]

Kualitas dari informasi tergantung dari 3 (tiga) hal, yaitu:

a. Akurat (*accurate*)

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

b. Tepat Pada Waktu (*time lines*)

Berarti informasi yang datang pada penerima tidak terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya informasi tersebut di dapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, pengolahan dan pengirimannya.

c. Relevan

Berarti informasi tersebut mempunyai manfaat untuk pemakainya relevan antar orang satu dengan orang lain berbeda-beda.

2.6 Konsep Dasar Sistem Pendukung Keputusan

Decision Support System (DSS) atau lebih dikenal dengan sistem pendukung keputusan merupakan salah satu produk perangkat lunak yang dikembangkan secara khusus untuk membantu manajemen dalam proses pengambilan keputusan. Sesuai dengan namanya tujuan digunakannya sistem ini adalah sebagai "*second opinion*" atau "*information source*" yang dapat dipakai sebagai bahan pertimbangan sebelum seorang manajer memutuskan kebijakan tertentu[7].

Pengambilan keputusan merupakan proses pemilihan alternative tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses

pengumpulan data menjadi informasi serta ditambah dengan faktor-faktor yang perlu dipertimbangkan dalam pengambilan keputusan.

2.6.1 Tahap-tahap Pengambilan Keputusan

Menurut Herbert A. Simon (Kadarsah, 2002:15-16), tahap-tahap yang harus dilalui dalam proses pengambilan keputusan sebagai berikut:

1. Tahap Pemahaman (*Intelligence Phace*)

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. Tahap Perancangan (*Design Phace*)

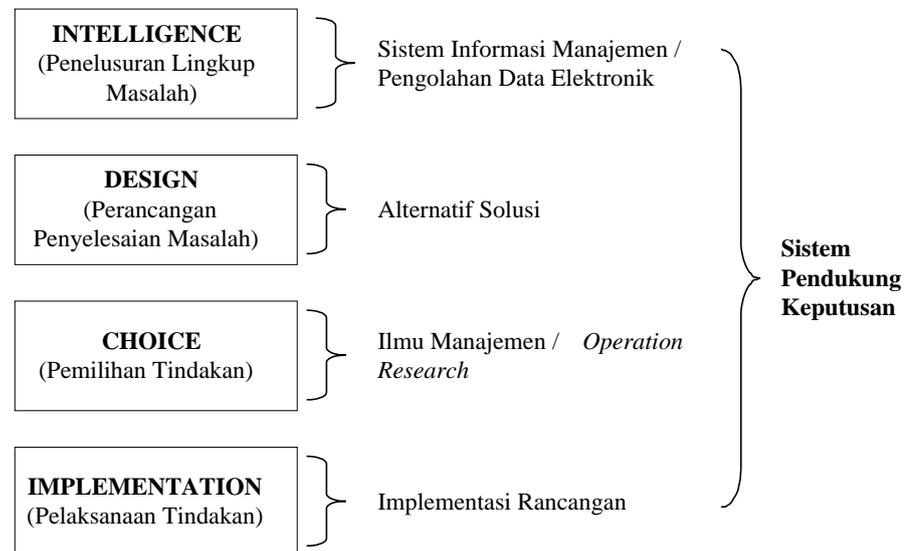
Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan / solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan (*Choice Phace*)

Tahap ini dilakukan pemilihan terhadap diantaraberbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan / dengan memperhatikan kriteria-kriteria berdasarkan tujuan yang akan dicapai.

4. Tahap Impelementasi (*Implementation Phace*)

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan.



Gambar 2.2 : *Fase Proses Pengambilan Keputusan* [8]

2.6.2 Jenis-Jenis Keputusan

Keputusan-keputusan yang dibuat pada dasarnya dikelompokkan dalam 2 jenis, antara lain :

1. Keputusan Terprogram

Keputusan ini bersifat berulang dan rutin, sedemikian hingga suatu prosedur pasti telah dibuat menanganinya sehingga keputusan tersebut tidak perlu diperlakukan de novo (sebagai sesuatu yang baru) tiap kali terjadi.

2. Keputusan Tak Terprogram

Keputusan ini bersifat baru, tidak terstruktur dan jarang konsekuen. Tidak ada metode yang pasti untuk menangani masalah ini karena belum ada sebelumnya atau karena sifat dan struktur persisnya tak terlihat atau rumit atau karena begitu pentingnya sehingga memerlukan perlakuan yang sangat khusus.

2.6.3 Tujuan Sistem Pendukung Keputusan

- a. Membantu manajer membuat keputusan untuk memecahkan masalah semi terstruktur
- b. Mendukung penilaian manajer bukan mencoba menggantikannya
- c. Meningkatkan efektivitas pengambilan keputusan manajer dari pada efisiensinya

Tujuan dari sistem pendukung keputusan bukanlah untuk membuat proses pengambilan keputusan seefisien mungkin tetapi manfaat utama menggunakan sistem pendukung keputusan adalah keputusan yang lebih baik.

2.7 Konsep Dasar AHP (*Analytic Hierarchy Process*)

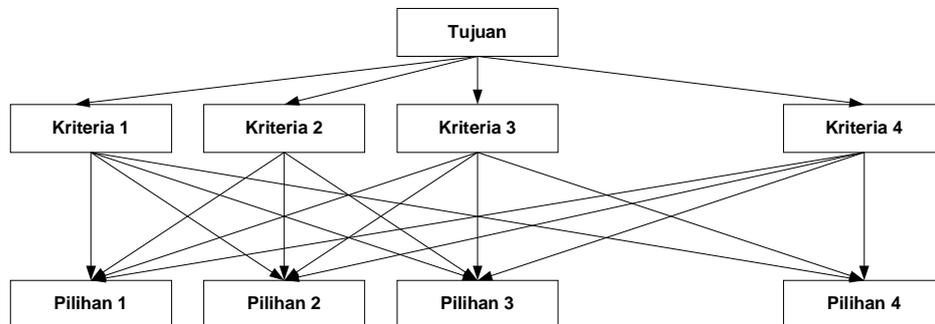
Proses pengambilan Keputusan pada dasarnya adalah memilih suatu alternatif. Peralatan utama *Analytic Hierarchy Process* (AHP) adalah sebuah hirarki fungsional dengan input utamanya persepsi manusia. Dengan hirarki, suatu masalah kompleks dan tidak terstruktur dipecahkan ke dalam kelompok-kelompoknya, kemudian kelompok-kelompok tersebut diatur menjadi suatu bentuk hirariki. (Permadi, 2002)

2.7.1 Prinsip Dasar AHP (*Analytic Hierarchy Process*)

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah :

1. Membuat hierarki

Sistem yang kompleks bisa dipahami dengan memecahkannya menjadi elemen-elemen pendukung, menyusun elemen secara hierarki dan menggabungkannya atau mensistensinya.



Gambar 2.3 : Struktur Hierarki dalam AHP

2. Penilaian

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty (1998), untuk berbagai persolah, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan defenisi pendapat kualitatif dari skala perbandingan Saaty bisa diukur menggunakan tabel analisis seperti ditunjukkan pada tabel berikut :

Tabel 2.1 : Skala penilaian Perbandingan Pasangan

Intensitas Keperingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen lainnya
5	Elemen yang satu lebih penting daripada elemen lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Jika aktivitas mendapat satu angka dibandingkan dengan aktivitas j, maka i memiliki nilai kebalikannya dibandingkan dengan i

Sumber : [9]

3. *Synthesis Of Priority* (menentukan prioritas)

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*Pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan *judgment* yang telah ditentukan untuk menghasilkan bobot dan prioritas.

4. *Logical Consistency* (Konsistensi Logis)

Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi, kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu.

2.7.2 Prosedur AHP

Pada dasarnya, prosedur atau langkah-langkah dalam metode AHP meliputi :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi.
2. Menentukan prioritas, meliputi menentukan prioritas elemen dan matriks perbandingan berpasangan diisi menggunakan bilangan untuk mempresentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.
3. Sistesis, terdiri dari menjumlahkan nilai-nilai dari setiap kolom pada matriks, membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks dan menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.
4. Mengukur konsistensi, dengan cara mengalikan setiap nilai pada kolom pertama dengan prioritas elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua dan seterusnya.

5. Menghitung *consistency index* (CI) dengan rumus

$$CI = (\lambda \text{ maks} - n) / n$$

Dimana n adalah banyaknya elemen

6. Hitung Rasio Konsistensi/*Consistency Ratio* (CR) dengan rumus :

$$CR = CI/RC$$

Dimana CR = *Consistency Ratio*

CI = *Consistency Index*

IR = *Index Random Consistency*

7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian data judgment harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar.

Daftar Indeks Random Konsistensi (IR) bisa dilihat dalam tabel berikut :

Tabel 2.2 Daftar Indeks Random Konsistensi

Ukuran Matriks	Nilai IR
1,2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,48
13	1,56
14	1,57
15	1,59

Sumber : [9]

2.8 Definisi *Software Engineering* (Rancang-Bangun Perangkat Lunak)

Definisi awal dari rancang-bangun perangkat lunak diutarakan oleh Fritz Bauer dalam suatu konferensi, dimana pada saat itu dijadikan sebagai pokok materi yang akan dibahas. Definisinya adalah :

The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.(Roger S Pressman, *Software Engineering A Practitioner's Approach*: Hal 19: Singapore 1987)

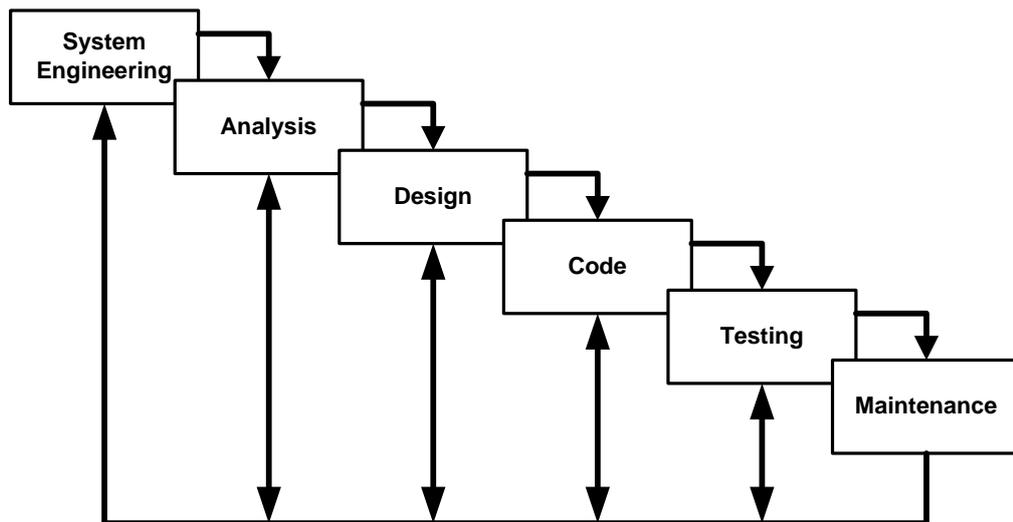
Dari sekian banyak definisi yang telah diusulkan, semua definisi tersebut mengacu pada pentingnya rancang-bangun dalam pengembangan software. Rancang-bangun perangkat lunak merupakan suatu pertumbuhan diluar perangkat keras dan rancang-bangun sistem. yang meliputi tiga kunci pokok yaitu : cara, alat dan prosedur yang memungkinkan manajer untuk mengendalikan proses pengembangan *software* dan menyediakan praktisi dengan tujuan untuk membangun perangkat lunak yang berkualitas.

Metoda rancang-bangun perangkat lunak menyediakan cara teknis "bagaimana cara" untuk membangun perangkat lunak, caranya meliputi suatu rangkaian tugas yang meliputi: perencanaan proyek dan penilaian; analisa system dan software yang dibutuhkan; perancangan struktur data, perancangan program, dan algoritma program; pengkodean; pengujian; dan pemeliharaan. Metoda untuk rancang-bangun perangkat lunak sering memperkenalkan suatu notasi grafis atau berorientasi bahasa khusus dan satu set kriteria untuk perangkat lunak yang berkualitas.

2.9 *System Development Life Cycle* (SDLC)

Gambar 2.1 menggambarkan paradigma Siklus Hidup Klasik (*The Classic Life Cycle*) untuk rancang-bangun perangkat lunak (*Software Engineering*), yang lebih dikenal dengan nama "Model air terjun". Paradigma Siklus hidup klasik menuntut sesuatu yang sistematis, yang mendekati contoh

pengembangan software yang dimulai pada tingkat sistem sampai pada analisa, disain, pengkodean, pengujian dan pemeliharaan, Dalam siklus rancang-bangun yang konvensional memiliki tahapan yang meliputi :



Gambar 2.4 : *The Classic Life Cycle* [10]

1. Rancang-Bangun Sistem (*System Engineering*) dan Analisa.

Karena perangkat lunak selalu merupakan bagian dari suatu sistem yang besar, pada tahap ini dimulai dengan penentuan kebutuhan untuk semua unsure-unsur sistem dan kemudian membagi menjadi beberapa subset dari kebutuhan ini yang salah satunya ke dalam perangkat lunak. Gambaran Sistem ini dibutuhkan apabila perangkat lunak harus berhubungan dengan unsur-unsur lain seperti perangkat keras, orang-orang dan data base. Rancang-bangun sistem dan analisa meliputi kebutuhan yang dikumpulkan pada tingkat sistem yang lebih rendah dari Top-Level desain dan analisa.

2. Analisa Kebutuhan Perangkat Lunak.

Proses pengumpulan kebutuhan diintensifkan dan secara khusus terpusat pada perangkat lunak. Untuk memahami sifat alamiah program

dalam pembuatannya, *software engineer* ("analisis") harus memahami informasi tentang perangkat lunak, seperti halnya fungsi yang akan dijalankan dan kemampuannya. Kebutuhan dari sistem dan perangkat lunak didokumentasikan dan ditinjau bersama dengan pelanggan.

3. Disain

Disain perangkat lunak benar-benar suatu proses yang mempunyai banyak tahapan yang berfokus pada 3 atribut program, yaitu : Struktur data, Arsitektur perangkat lunak dan Mengenai cara yang lebih mendetail. Proses disain menterjemahkan kebutuhan ke dalam suatu presentasi perangkat lunak yang dapat digunakan sebagai penilaian kualitas sebelum memulai pengkodean.

4. Pengkodean (*Coding*)

Disain harus bisa diterjemahkan ke dalam suatu format yang terbaca oleh mesin. Langkah pengkodean yang dilaksanakan pada bagian ini. Jika disain dilakukan dalam suatu cara yang terperinci, pengkodean dapat terpenuhi secara mekanistik.

5. Pengujian (*Testing*)

Tahap ini bisa dilakukan hanya apabila proses pengkodean telah selesai. Proses pengujian memusatkan pada logika internal dari perangkat lunak, meyakinkan bahwa semua statemen telah diuji, dan pada fungsional eksternal yaitu melaksanakan test untuk meyakinkan masukan yang digambarkan itu akan menghasilkan keluaran yang nyata yang disepakati sebagai hasil telah diminta.

6. Pemeliharaan (*Maintenance*)

Perangkat lunak lambat laun niscaya akan mengalami perubahan setelah digunakan oleh pelanggan (suatu perkecualian mungkin penambahan perangkat lunak). Perubahan akan terjadi bisa disebabkan oleh perangkat lunak harus menyesuaikan diri untuk mengakomodasi perubahan dalam lingkungan eksternalnya (misalnya,, suatu perubahan diperlukan oleh karena

sistem operasi atau perangkat keras yang digunakan telah berbeda dan lebih maju), atau disebabkan oleh keperluan fungsional pelanggan atau peningkatan kemampuan software. Pemeliharaan perangkat lunak berlaku untuk semua tahapan dalam siklus kehidupan untuk program yang telah ada.

2.10 Alat Bantu Perancangan

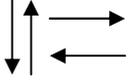
2.10.1 Sistem Prosedur Diagram Secara Manual

Digunakan untuk membuat *flow of document* (manual). Fungsi diagram ini untuk mendefinisikan hubungan antar bagian (pelaku proses), proses manual dan aliran data (dalam bentuk dokumen masukan dan keluaran).

Adapun simbol-simbol yang digunakan dalam *Flow Of Document* adalah :

Tabel 2.3 Simbol-simbol *Flow Of Document*

Simbol	Keterangan
	DOKUMEN, digunakan untuk mendefinisikan dokumen masukan (formulir) dan dokumen keluaran (laporan).
	PEMASUKAN DATA, digunakan untuk mendefinisikan pemasukan data (umumnya melalui keyboard), tetapi dapat juga masukan lain seperti digizer, mouse dan lain-lain.
	PEMASUKAN MANUAL, digunakan untuk mendefinisikan pekerjaan manual.
	ARSIP/DOKUMENTASI, digunakan untuk mendefinisikan penyimpanan arsip seandainya suatu saat diperlukan sebagai back-up, pembuatan bahan laporan, bahan audit dan sebagainya.
	PENGHUBUNG/KONEKTOR, digunakan untuk mendefinisikan penghubung kebagian lain tetapi masih dalam halaman yang sama
	PENGHUBUNG/KONEKTOR, digunakan untuk mendefinisikan penghubung kebagian lain di

	halaman yang berbeda
	GARIS ALIR, menunjukkan alur dari proses.

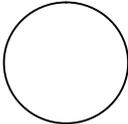
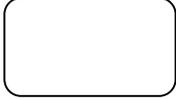
Sumber : [6]

2.10.2 Data Flow Diagram (DFD) atau Diagram Arus Data (DAD)

DFD adalah diagram dari aliran data melalui sebuah sistem. DFD dapat di bagi menjadi dua yaitu DFD fisik dan DFD logis. DFD digunakan untuk komunikasi antara analis dan pemakai. Tujuan penggunaan DFD untuk menghubungkan aliran data dari seluruh sistem.

Simbol-simbol yang di gunakan dalam DFD antara lain :

Tabel 2.4 Simbol *Data Flow Diagram*

Simbol	Fungsi
	PROSES Digunakan untuk menunjukkan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran ataupun sebaliknya.
	
 	ALIRAN DATA Digunakan untuk menggambarkan gerakan paket data atau informasi dari satu bagian ke bagian lain dari sistem dimana penyimpanan mewakili bakal penyimpanan data.
	PENYIMPANAN Dapat digunakan untuk mendefinisikan file atau basis data atau sering kali mendefinisikan bagaimana penyimpanan diimplementasikan dalam sistem komputer.

	<p>TERMINATOR (ASAL / TUJUAN DATA)</p> <p>Melambangkan orang atau kelompok orang (misalnya organisasi diluar sistem, grup, departemen, perusahaan, perusahaan pemerintah) yang merupakan asal data atau tujuan informasi</p>
---	---

Sumber : [6]

Diagram arus data itu sendiri dibagi menjadi dua bagian yaitu :

a. DFD Diagram Context

Merupakan alat untuk menjelaskan struktur analisis. Pendekatan ini mencoba untuk menggambarkan sistem pertama kali secara garis besar dan memecahkannya menjadi bagian yang terinci yang disebut dengan lower level. Dan yang pertama kali digambar adalah level yang teratas sehingga disebut Diagram Context.

b. DFD levelled

Setelah context diagram dirancang kemudian akan digambar lebih terinci lagi yang disebut dengan *over view* diagram (*level 0*). Tiap-tiap proses di *over view* diagram

akan digambar lebih terinci lagi dan disebut dengan *level 1*, dan kemudian di teruskan ke *level* berikutnya sampai tiap-tiap proses tidak dapat digambar lebih terinci.

2.10.3 Kamus Data (Data Dictionary)

Merupakan teknik lain untuk model data dalam sistem informasi dan merupakan tempat penyimpanan untuk semua *level* sederhana struktur data dan elemen data dalam sistem [10]. Dengan demikian kamus data dapat

mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap perancangan sistem kamus data digunakan untuk merancang *input*, merancang laporan-laporan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD. Arus data di DFD secara lebih terinci dapat dilihat di kamus data. Dan untuk mendefinisikan struktur data yang ada di kamus data biasanya digunakan notasi-notasi yang menunjukkan informasi-informasi tambahan. Notasi-notasi tersebut berbentuk :

Tabel 2.5 Simbol *Data Dictionary*

Simbol	Uraian
=	Sama dengan diuraikan menjadi, terdiri dari, mendefinisikan, artinya.
+	Dan
()	Optional (boleh ada / tidak)
{ }	Pengulangan
[]	Memilih salah satu dari alternatif
**	Komentar
	Pemisah sejumlah alternatif pilihan antara simbol

Sumber : [6]

2.10.4 Merancang *Database*

Merancang *database* mempunyai tujuan yaitu meminimumkan pengulangan data dan independensi data [13] Perancangan *database* diperlukan untuk menghindari permasalahan di dalam *database*.

2.10.4.1 ERD (*Entity Relationship Diagram*)

ERD adalah merupakan suatu model data untuk mengilustrasikan desain logika dari skema *database*. ERD terdiri dari tiga bagian :

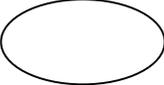
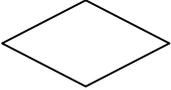
1. Entitas, yaitu suatu objek yang terdiri dari kumpulan data dari *database*.

2. Relasi, yaitu pengukur antar entitas
3. Atribut, yaitu menggambarkan hubungan antara entitas dan relasi.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data.

Simbol-simbol yang digunakan yaitu :

Tabel 2.6 Simbol-Simbol ERD

Simbol	Fungsi
<i>Entity</i> 	Digunakan untuk menggambarkan obyek yang dapat didefinisikan dalam lingkungan pemakai sistem.
Atribut 	Digunakan untuk menggambarkan elemen-elemen dari suatu entity, yang menggambarkan karakter entity.
Hubungan 	<i>Entity</i> dapat berhubungan satu dengan yang lain. Hubungan ini disebut <i>Relationship</i> .
Garis 	Digunakan untuk menghubungkan <i>entity</i> dengan <i>entity</i> dan <i>entity</i> dengan atribut.

Sumber : [13]

2.10.4.2 Normalisasi

Normalisasi adalah proses yang berkaitan dengan model data relasional untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi atau erat.[13].

Bentuk-bentuk Normalisasi :

1. Bentuk tidak normal

Merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikat suatu format tertentu, dapat saja tidak lengkap atau

terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk Normal Kesatu

Bentuk normal kesatu mempunyai ciri yaitu setiap data dibentuk dalam *flat file*. Data dibentuk dalam satu *record* dan nilai dari *field* berupa *atomik value*.

3. Bentuk Normal Kedua

Syarat yaitu bentuk data telah memenuhi kriteria kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama.

4. Bentuk Normal Ketiga

Syarat yaitu harus memenuhi bentuk normal kedua. Setiap atribut bukan kunci haruslah bergantung hanya pada kunci utama dan pada kunci utama secara menyeluruh.

2.10.5 Perancangan Masukan dan Keluaran (*Input Output Design*)

- **Desain Input**

Untuk membuat laporan baru ke dalam perkembangan sistem adalah dengan menggunakan dokumen dan prosedur desain input sehingga perkembangan sistem dapat berkembang dengan cepat dan akurat pada sistem informasi.[10].

Tujuan desain input :

1. Untuk mengefektifkan biaya pemasukan data.
2. Untuk menjamin pemasukan data dapat diterima dan dimengerti oleh pemakai.

- **Desain Output**

Desain output merupakan keberhasilan dari sistem informasi. Setelah desain input dan file berlangsung maka akan terjadi desain output [10].

2.11 Metode Pengujian Aplikasi

Pengujian perangkat lunak merupakan proses eksekusi program atau perangkat lunak dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi atribut dan kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau output yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Ada berbagai macam metode pengujian, teknik black box dan teknik white box merupakan metode pengujian yang telah dikenal dan banyak digunakan oleh pengembang perangkat lunak.

2.11.1 Metode Pengujian *Black Box*

Metode pengujian *black box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*). Pendekatan ini hanya mengevaluasi program dari *output* atau hasil akhir yang dikeluarkan oleh program tersebut. Struktur program dan kode-kode yang ada di dalamnya tidak termasuk dalam pengujian ini. Keuntungan dari metode pengujian ini adalah murah dan sederhana. Namun, pengujian dengan metode ini tidak dapat mendeteksi kekurangefektifan pengkodean dalam suatu program.

2.11.2 Metode Pengujian *White Box*

Metode pengujian *white box* atau dapat disebut juga glass box merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah perangkat lunak atau program sebagai kotak kaca (*glass box*). Pendekatan ini akan mengevaluasi stuktur program dan kodenya yang meliputi efektivitas pengkodean, pernyataan kondisional (alur program), dan looping yang digunakan dalam program. Keuntungan dari metode pengujian ini antara lain dapat ditemukannya kode-kode tersembunyi yang menghasilkan kesalahan serta dapat menghasilkan program yang efektif.

2.12 Borland Delphi 7.0

Dasar bahasa pemrograman yang digunakan dalam Delphi adalah Pascal, sebuah bahasa yang didesain khusus oleh Niklaus Wirth untuk mengajarkan pemrograman terstruktur. Dibandingkan dengan bahasa generasi ketiga lainnya, seperti bahasa C, Pascal lebih mudah dipelajari dan digunakan. Hal ini karena Pascal memiliki struktur bahasa seperti bahasa Inggris sehingga mudah untuk dibaca. Tipe data dalam pascal antara lain adalah : *Integer, Real, Boolean, Char, String, Pointer, Pchar*.

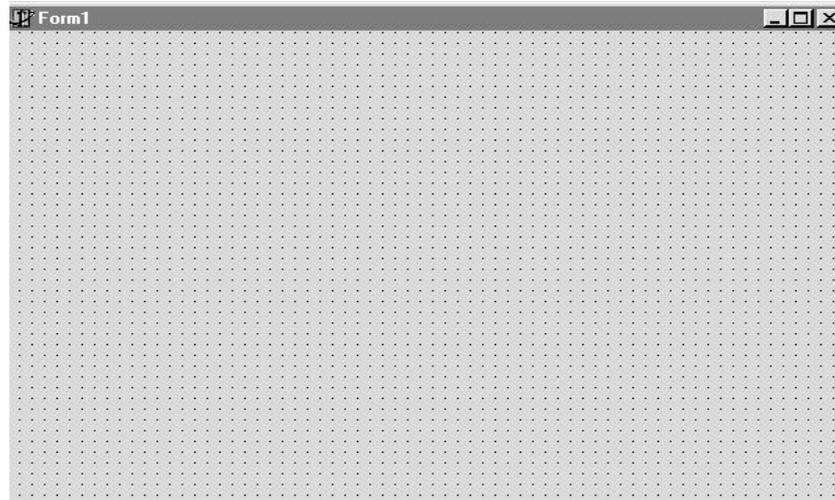
Pascal dalam Delphi berbeda dengan Pascal pada versi-versi sebelumnya, bahkan bila dibandingkan dengan Borland Pascal Versi 7.0, objek Pascal dalam Pascal 7 merupakan pengembangan kompiler-kompiler Pascal versi sebelumnya. Sekarang dalam Delphi bentuk objek Pascal lebih ditingkatkan lagi oleh Borland. Delphi dikembangkan dengan tujuan untuk menetapkan standar baru bahasa Pascal. Akan tetapi Delphi masih mampu mengenal bentuk-bentuk lama objek Pascal dari versi kompiler yang lama.

Bahasa pascal adalah bahasa yang *strongly-typed*. Artinya, variabel ini harus selalu diberi nilai yang tipenya sama dengan deklarasinya. Bila tidak maka kompiler akan memberikan pesan kesalahan. Sedangkan bahasa yang *weak-typed* seperti bahasa C, variabelnya dapat menerima nilai yang berbeda dengan type deklarasinya. [12]

2.12.1 Form Dan Editor Program

Windows Form dan *Windows Editor* program berhubungan sangat erat. Istilah Unit dalam Delphi adalah sebuah file program sumber (file PAS) untuk suatu form.

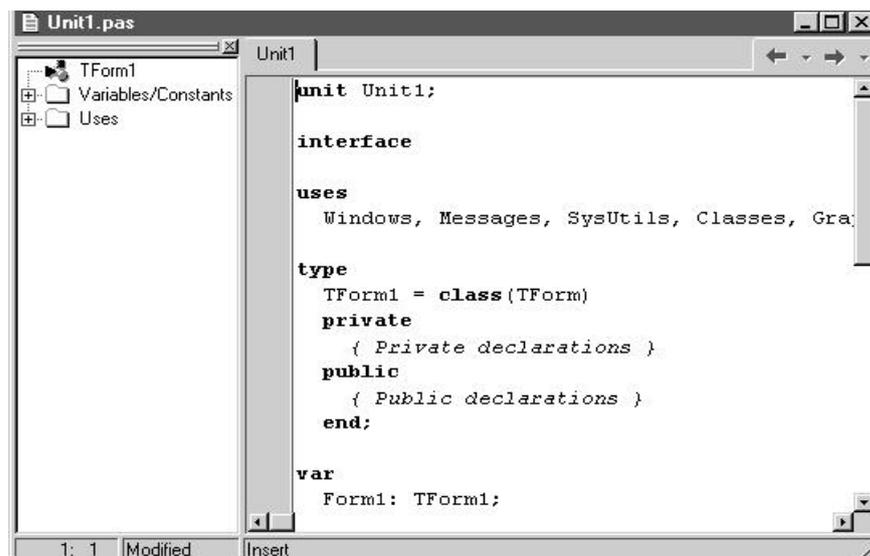
Window form adalah tempat dimana semua komponen visual dan komponen non visual diletakkan untuk membentuk sebuah aplikasi.



Gambar 2.5 : *Window Form* Pada Delphi

Sumber : [10]

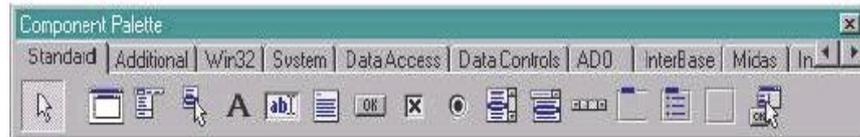
Sedangkan pada saat program *editor* aktif, akan dilengkapi fasilitas *editor* yang berdaya guna. Cara kerja *editor* ini sama dengan aplikasi *Brief*, sebuah *editor* yang sangat populer dikalangan para pemrogram.



Gambar 2.6 : *Window Editor* Pada Delphi

Sumber : [10]

2.12.2 Komponen

Gambar 2.7: *Component Palette*

Sumber : [10]

Dalam membuat program, Delphi telah menyediakan banyak kemudahan, yaitu dengan disediakannya komponen-komponen. Komponen ini merupakan sebuah *procedure*/program yang sudah di compile dan langsung dapat digunakan, sesuai dengan fungsinya masing-masing. Untuk menggunakan komponen ini kita dapat meng-klik komponen yang diinginkan, kemudian kita klik di form, maka komponen tersebut akan muncul di form.

Kode yang akan dilihat, serupa dengan struktur Bahasa Pascal. Delphi adalah bahasa pemrograman berbasis objek, artinya pendekatan pembuatan program melalui objek-objek yang ada. Misalnya objek form, text dsb. Setiap objek akan memiliki properti (atribut) dan method yang diaktifkan / dipicu oleh *event*.

Kegunaan beberapa komponen :

a. **Button/ Bitbtn**

Biasa digunakan sebagai tombol kendali. Perbedaan antara *bitbtn* dengan *button*: pada *bitbtn* kita dapat menyisipkan warna pada tombol dan icon tertentu, lain halnya bila kita menggunakan *button*.

b. Panel

Panel berfungsi untuk mengelompokan komponen-komponen didalamnya.

c. Label

Kita dapat menamakan atau memberi keterangan pada program

d. Edit

Edit berfungsi sebagai masukan data (*input*) dalam bentuk *string*, dari bentuk *string* ini kita dapat mengolahnya menjadi bentuk *integer* atau bentuk lainnya yang kemudian dapat digunakan untuk operasi selanjutnya.

e. Chart

Data-data yang telah kita analisa, dapat kita tampilkan ke dalam grafik, sehingga memudahkan kita untuk menganalisanya.

f. Stringgrid

Stringgrid berguna untuk menaruh data *string* kedalam bentuk kolom tabel, seperti pada Excel. Kita harus mengubah *type* data ke dalam bentuk *string* bila data yang ingin kita tampilkan data bukan *string*.

g. Popup Menu

Popup Menu berfungsi sebagai perintah yang aktif bila kita mengklik kanan *mouse*, Untuk mengaktifkannya kita harus mengaktifkan *Popup Menu* pada komponen yang diinginkan, caranya : ubah pada *object inspector*.

h. MainMenu

Contoh *Main Menu* adalah *Option* pada tiap aplikasi program, dengan komponen ini, kita bisa menaruh fungsi-fungsi program seperti pada aplikasi umumnya.

i. ComboBox

Combo Box berfungsi sebagai petunjuk untuk pemilihan berbagai masukan.

j. *CheckBox*

Bila komponen ini di *check* maka ada aplikasi yang bisa disetting untuk bekerja dibawahnya.

k. *RadioButton*

Prinsip kerjanya hampir sama dengan *check box*, cuma tampilannya saja yang berbeda.

l. *Media Player*

Biasa digunakan untuk menyalakan atau memainkan musik (format *wav* atau *midi*) dan menjalankan film (format *avi*).

m. *Timer*

Timer berfungsi sebagai jam yang telah disediakan Delphi. Dengan *timer* kita juga dapat mendecode time, sehingga dapat terjadi akusisi data.

n. *Clientsocket*

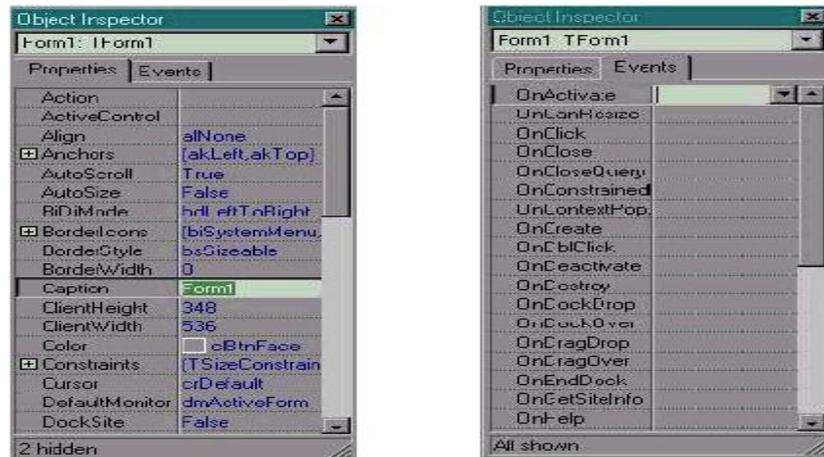
Digunakan di *client* pada saat koneksi ke *server*.

o. *Serversocket*

Digunakan di *server*, yang akan menyediakan layanan koneksi ke *client*. Masih banyak komponen lain. Untuk mempelajari lebih lanjut dapat dibaca pada *help menu*

2.12.3 *Object Inspector*

Object inspector (gambar dibawah) berguna sebagai *options* dari masing-masing komponen. Dengan *object inspector* ini kita dapat memanipulasi komponen yang kita gunakan (walaupun sebenarnya kita juga dapat menggunakannya dengan menuliskannya lewat *text mode*).



Gambar 2.8 : Object Inspektor

Sumber : [10]

Beberapa contoh pada *Properties* :

a. Font

Menunjukkan *font* yang akan kita gunakan. Dalam menu ini juga terdapat macam-macam *font* berikut *style* dan ukurannya.

b. Caption

Merupakan judul dari *form* itu. Perhatikan perbedaannya dengan *name*.

c. Name

Merupakan nama dari komponen itu. Biasanya bila kita mengambil sebuah komponen, Delphi otomatis menyediakan nama sesuai dengan nama komponen tersebut. Namun nama ini dapat dengan mudah kita ubah agar mudah mengingatnya. Berbeda dengan *caption* pada *caption* kita tidak mengubah nama komponen, melainkan hanya mengubah judul yang tertera pada komponen tersebut. Sehingga bila kita ingin menggunakan komponen tersebut, yang kita gunakan adalah namanya, dan bukan nama pada *caption*.

d. Enabled

Menunjukkan bahwa komponen ini bisa digunakan atau tidak. Perintah ini cukup penting bila kita tidak ingin *user* menekan tombol yang tidak diinginkan.

e. Height

Merupakan tinggi dari komponen yang bakalan kita taruh pada *form*, atau komponen lainnya.

f. Width

Merupakan lebar komoponen.

g. Popup Menu

Fungsi ini berguna saat kita mengkilik kanan dan akan tampil perintah. Pada penggunaanya kita mengambil komponen *popup menu* dan menset *popup menu* pada *form*.

h. Auto Size

Fungsi ini berisi mengenai apakah *form* yang kita miliki bisa diperbesar atau tidak.

i. Text

Biasanya tedapat pada komponen edit dan masuk edit. Kegunaanya untuk menaruh kata-kata pada komonen ini.

j. Color

Warna pada komponen. Kita dapat mengubah warna-warna tiap komponen (hanya yang disediakan) dengan *point* ini.

k. Visible

Berfungsi untuk menampilkan dan tidak menampilkan komponen yang kita inginkan.

l. Hint

Bila kursor kita diatas komponen, maka akan muncul suatu keterangan. Keterangan ini yang disebut dengan *hint*. Jangan lupa

untuk mengganti properti *show hint* menjadi *true* untuk mengaktifkannya.

m. *BorderStyle*

Menunjukkan berbagai macam bingkai yang diinginkan. Masih banyak properti lainnya, bisa dipelajari lebih lanjut.

Dengan *Object inspector* kita juga dapat menentukan *event* apa yang akan kita buat dengan komponen tersebut.

Beberapa contoh *Event*.

a. *OnClick*

Bila kita mengklik komponen tersebut maka prosedur yang kita inginkan akan dijalankan oleh program tersebut.

b. *Onkeypress*

Bila kita menekan suatu *key* (tombol) maka komponen tersebut akan aktif perlu diperhatikan pada *event* ini, Delphi mengenal *character* yang kita tekan melalui kode yang masuk kedalam variabel *key*.

Lihat contoh ini.

```
Procedure TForm1.Edit1KeyPress(Sender: TObject; var Key:
Char);
begin
if key =#13 then .....(isikan perintah)
end;
```

Bila key =#13 (enter) maka perintah yang ikuti akan dijalankan.

2.12.4 File-File Pada Delphi

b. **.pas*

Merupakan *source file*, disini akan disimpan kode pascal yang kita tulis.

c. *.dpr

Merupakan *project file*. Sebagai *project file*, *file* ini berguna untuk menggabungkan satu atau lebih *file-file source* (*.pas).

d. *.dcu (Delphi Compiled Unit)

Pada saat kita membuat sebuah aplikasi, Delphi akan membuat *file* ini. *File* ini berfungsi untuk me-*link*-kan kita dengan *file* lain (*.dcu) sehingga kita bisa membuat beberapa *form* yang terhubung satu dengan yang lainnya.

e. *.dfm (Delphi Form)

File ini berisi informasi mengenai data-data *form*.

f. *.dof (Delphi Project Options)

Kita dapat mengubah *icon* aplikasi tersebut, dan datanya disimpan pada *file* ini.

g. *.exe

Merupakan *application file* setelah kita *compile* program kita.

2.12.5 Type Data Pada Delphi

Pada umumnya Delphi menggunakan type data yang sama dengan pascal. Namun type data pada Delphi lebih lengkap dibanding pascal. Tipe data yang biasa digunakan pada Delphi :

a. String

Biasa digunakan untuk masukan data *string* (*String* merupakan gabungan beberapa *character*).

b. Integer

Merupakan bilangan asli (1, 2, 3, 4, ...).

c. Real

Bilangan bulat (berupa pecahan dan *integer*).

d. Byte

Besar dari type ini hanya 8 *bit*.

e. Word

Besar bilangan ini 16 *bit*.

f. *Date And Time*

Type waktu dan tanggal yang disediakan Delphi.

2.12.6 Konversi Type Data

Dalam pengolahan data biasanya kita memerlukan suatu konversi type data, banyak sekali konversi data yang bisa dilakukan pada Delphi.

Contoh konversi type data :

a. *Strtoint*

Mengubah type data *string* ke *integer*.

b. *Inttostr*

Mengubah type data *integer* ke *string*.

c. *Timetostr*

Mengubah type data *time* ke dalam bentuk *string*.

d. *Strtfloat*

Mengubah type data *string* ke dalam bentuk *real*.

e. *Floattostr*

Mengubah type data *real* ke dalam bentuk *string*.