

BAB II

KAJIAN PUSTAKA

II.1 Penelitian Terkait

Penelitian terdahulu merupakan referensi bagi penelitian untuk melakukan penelitian ini.

1. Septiyana Firdyana, dkk, Vol. 2 No. 1 Maret 2017, ISSN: 2540-7902. Penerapan Metode *Weighted Product* Untuk Menentukan Penerima Bantuan Beras Masyarakat Miskin (Raskin). Hasil jurnal ialah bahwa memberikan bantuan kepada masyarakat miskin dalam kajian ini digunakan berbagai macam kriteria yang telah ditentukan sebelumnya oleh pihak pemerintah dengan metode *Weighted Product* agar dapat menyeleksi masyarakat yang berhak mendapatkan bantuan beras.
2. Ratna Fitria dan Syifa Nur Rakhmah, ISSN : 2085-0859 Vol. 11 Nomor 2 Tahun 2019. Penerapan Metode *Weighted Product* Untuk Menganalisis Faktor Pemilihan Pembayaran *Digital Transaction* Pada Aplikasi Ojek *Online*. Hasil jurnal ialah menentukan pembayaran digital terbaik untuk aplikasi ojek online berdasarkan kriteria-kriteria tertentu. Dimana hasil ini dapat digunakan sebagai acuan dalam memilih pembayaran digital para masyarakat dalam bertransaksi dengan ojek online.
3. Nur Aini dan Fahrul Agus. Vol. 9, No. 2, Mei 2017. Penerapan Metode *Weighted Product* Dan *Analytic Hierarchy Process* Untuk Pemilihan Koperasi Berprestasi. Hasil jurnal ialah untuk memilih koperasi terbaik di kota

Samarinda. Dengan pemilihan berdasarkan beberapa kriteria ditentukan oleh dinas koperasi dan UKM kota Samarinda.

4. Noprin Pakaya dan Amiruddin. Vol 9, No 3, April 2017 ISSN 2087-1716. Sistem Pendukung Keputusan Pemilihan Peserta Kapal Pemuda Nusantara Dengan Menggunakan Metode *Weighted Product* (WP). Hasil jurnal ialah untuk menentukan yang berhak menjadi peserta kapal pemuda nusantara berdasarkan kriteria dengan menggunakan rumus yang hasilnya lebih tepat dan lebih akurat.
5. Muhammad Sapruwan. Vol. 1 No. 1, Januari 2020, ISSN : 2716-0238. Penerapan Metode *Weighted Product* Dalam Proses Seleksi Calon Karyawan di Perusahaan Perkebunan Kelapa Sawit. Hasil jurnal ialah untuk membantu perusahaan dalam pengambilan keputusan menentukan calon karyawan yang layak diterima dengan lebih cepat dan lebih akurat.

II.2 Sistem Pendukung Keputusan

Michael Scoot Morton pertama kali mencetuskan Sistem Pendukung Keputusan dengan istilah *Management Decision System*, yang tujuannya untuk mendukung manajemen pengambilan keputusan. Sistem pendukung keputusan adalah sistem berbasis komputer yang dibangun lewat sebuah proses adaptif dari pembelajaran, pola-pola penggunaan dan evolusi sistem. Bonzcek mendefinisikan sistem pendukung keputusan sebagai sebuah sistem berbasis komputer yang terdiri atas beberapa komponen, antara lain komponen sistem bahasa (*language*)

dan komponen sistem pengetahuan (*problem processing*) yang saling berinteraksi satu dengan lainnya (Amelia Budi, 2019).

Sistem pendukung keputusan adalah sistem informasi berbasis komputer yang interaktif, dengan cara mengolah data dengan berbagai model untuk memecahkan masalah-masalah yang tidak terstruktur sehingga dapat memberikan informasi yang bisa digunakan oleh para pengambil keputusan dalam membuat sebuah keputusan. Dari definisi diatas dapat disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan semi terstruktur.

Adapun dukungan Sistem Pendukung Keputusan sebagai berikut (Sofian Wira Hadi, 2019):

1. Analisis Jika-Maka (*What-if-analysis*)

Analisa yang dititik beratkan pada pengobservasian perubahan terhadap variabel-variabel tertentu berpengaruh terhadap variabel lain.

2. Analisis Sensitivitas (*Sensitivity Analysis*)

Analisa yang dititik beratkan pada pengobservasian pengaruh naik turunnya suatu variabel terhadap variabel-variabel lainnya.

3. Analisis Pencarian Sasaran (*Goal Seeking Analysis*)

Analisa yang dititik beratkan pada perubahan nilai-nilai beberapa variable hingga nilai variable yang diinginkan mencapai nilai tertentu.

4. Analisis Optimisasi (*Optimization Analysis*)

Analisa yang dititik beratkan pada pencarian nilai-nilai optimum dari setiap variabel yang ada berdasarkan pada kendala-kendala yang ada.

II.2.1 Konsep Dasar Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan atau *Decision Support System* pertama kali dikemukakan oleh Michael S. Scott Morton pada tahun 1971 dengan istilah *Management Decision System*. Michael S.Scott Morton mendefinisikan Sistem Pendukung keputusan sebagai sistem berbasis komputer interaktif, yang membantu para pengambil keputusan untuk memanfaatkan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur.

Menurut Little Man dan Watso Sistem Pendukung Keputusan adalah suatu sistem yang interaktif, yang membantu pengambil keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur. Dari definisi yang dikemukakan diatas dapat disimpulkan bahwa Sistem Pendukung Keputusan adalah suatu sistem informasi interaktif yang ditujukan untuk membantu para pengambil keputusan dalam memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur (Doni Winarso, 2019).

II.2.2 Komponen Sistem Pendukung Keputusan

Sistem Pendukung Keputusan terdiri dari 4 komponen utama, yakni (Ichsan Fahmi, 2019):

1. Subsistem pengelolaan data (*database*).

Subsistem pengelolaan data merupakan komponen sistem pendukung keputusan penyedia data bagi sistem. Data tersimpan dalam suatu pangkalan data (*database*) yang diorganisasikan suatu sistem yang disebut sistem manajemen pangkalan data (*Data Base Manajemen System/DBMS*).

2. Subsistem pengelolaan model (*model base*).

Subsistem Manajemen Model mencakup elemen-elemen berikut:

- a. Model Dasar
- b. Sistem Manajemen Model Dasar
- c. Bahasa Pemodelan
- d. Direktori Model
- e. Model Eksekusi (*Execution Model*), Integrasi (*Integration*) dan Pemrosesan Komentar (*Command Processing*)

3. Subsistem pengelolaan dialog (*user interface*).

Keunikan lain dari sistem pendukung keputusan adalah adanya fasilitas yang mampu mengintegrasikan sistem terpasang dengan pengguna secara interaktif.

Fasilitas yang dimiliki oleh subsistem ini dibagi atas tiga komponen, yaitu:

- a. Bahasa aksi, yaitu suatu perangkat lunak yang dapat digunakan pengguna untuk berkomunikasi dengan sistem. Komunikasi dilakukan melalui berbagai pilihan media seperti *keyboard*, *joystick* dan *key function*.
- b. Bahasa Tampilan, yaitu suatu perangkat yang berfungsi sebagai sarana untuk menampilkan sesuatu.

c. Basis Pengetahuan, yaitu bagian yang mutlak diketahui oleh pengguna system dan dirancang dapat berfungsi secara efektif.

4. Subsistem manajemen berbasis pengetahuan

Subsistem ini mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen independen dan bersifat opsional. Selain memberikan intelegensi untuk memperluas pengetahuan si pengambil keputusan, subsistem ini bisa diinterkoneksi dengan repositori pengetahuan perusahaan (bagian dari sistem manajemen pengetahuan), yang kadang-kadang disebut basis pengetahuan opsional.

II.2.3 Tujuan Sistem Pendukung Keputusan

Di dalam Sistem Pendukung Keputusan terdapat tiga tujuan yang harus dicapai yaitu (Danang Arbian, 2017):

1. Membantu manajer dalam pembuatan keputusan untuk memecahkan masalah semi terstruktur.
2. Mendukung keputusan manajer, dan bukanya mengubah atau mengganti keputusan tersebut.
3. Meningkatkan efektifitas manajer dalam pembuatan keputusan dan bukannya peningkatan efisiensi.

II.3 *Weighted Product*

Metode *Weighted Product* merupakan metode untuk menyelesaikan *Multi Attribut Decision Making* (MADM). *Weighted Product* menggunakan teknik perkalian untuk menghubungkan rating atribut, dimana rating tiap atribut harus

dipangkatkan terlebih dahulu dengan bobot atribut yang bersangkutan (Septiana Fidyana, 2017).

Algoritma metode *weighted product* dilakukan dengan melakukan perkalian untuk menghubungkan rating atribut. Rating setiap atribut dipangkatkan dengan bobot yang bersangkutan. Proses ini mirip dengan proses normalisasi.

Preferensi untuk alternative S_i diberikan sebagai berikut:

$$S_i = \prod_{j=1}^n X_{ij} W_j$$

dengan $i = 1, 2, \dots, m$. dimana :

S : *Preferensi* alternatif dianalogikan sebagai vektor S

X : Nilai kriteria

W : Bobot kriteria/subkriteria

I : Alternatif

j : Kriteria

n : Banyaknya kriteria

Dimana $\sum W_j = 1$. W_j adalah pangkat bernilai positif untuk atribut keuntungan dan bernilai negatif untuk atribut biaya. Preferensi relatif dari setiap alternatif, diberikan sebagai :

$$V_i = \frac{\prod_{j=1}^n X_{ij} W_j}{\prod_{j=1}^n (X_j^*)}$$

dimana :

V : *Preferensi* alternatif dianalogikan sebagai vektor V

X : Nilai Kriteria

W : Bobot kriteria/subkriteria

i : Alternatif

j : Kriteria

n : Banyaknya kriteria

* : Banyaknya kriteria yang telah dinilai pada vektor S

II.4 *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) adalah sebuah bahasa berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan sebuah sistem pengembangan perangkat lunak berbasis objek. *Unified Modeling Language (UML)* memungkinkan melakukan pemetaan (*mapping*) langsung dari model-model yang dibuat dengan bahasa-bahasa pemrograman berorientasi obyek.

Pada penelitian ini menggunakan perancangan UML sebagai berikut (Mohammad Nur Fawaiq, 2019):

1. *Use Case Diagram*

Bagaimana user berinteraksi dengan sebuah sistem.

2. *Activity Diagram*

Perilaku prosedural dan paralel.

3. *Class Diagram*

Perilaku prosedural dan paralel.

II.4.1 Tujuan Penggunaan UML

Tujuan *Unified Modeling Language (UML)* adalah sebagai berikut di bawah ini [14]:

- a. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan yang dapat saling menukar model dengan mudah dan dimengerti secara umum.
- b. Memberikan bahasa pemodlan yang bebas dari berbagai bahasa pemrograman dan proses rekayaasa.
- c. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

II.4.2 Bagian-Bagian UML

Bagian-bagian utama dari UML adalah *view*, diagram, model *element*, dan *general mechanism*. Berikut penjelasannya *view*, diagram, model *element*, dan *general mechanism*:

1. *View*

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

2. Diagram

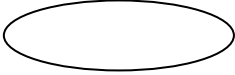
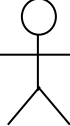

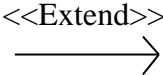
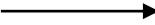
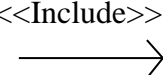
Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

a. *Use Case Diagram*

Use case diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use case*

mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel II.1 Simbol *Use Case Diagram*


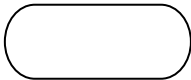
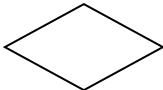
No	Simbol	Deskripsi	Keterangan
1		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor.
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat
3		<i>Association</i>	Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i>
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i>
5		<i>Generalitation</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini



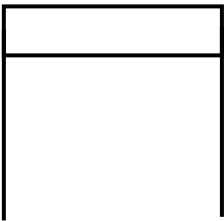
b. Activity Diagram

Activity diagram adalah diagram yang menggambarkan sifat dinamis secara alamiah sebuah sistem dalam bentuk model aliran dan kontrol dari aktivitas ke aktivitas lainnya (Nurul Azwanti, 2017).

Pada diagram ini digambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. Diagram *activity* digunakan untuk menggambarkan urutan aliran kegiatan-kegiatan dari sebuah proses bisnis atau sebuah *use case*. Diagram ini juga dapat digunakan untuk memodelkan aksi dan hasil ketika operasi berlangsung. Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram*:

Tabel II.2 Simbol-Simbol Activity Diagram

No	Simbol	Deskripsi	Keterangan
1		Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu



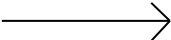
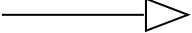
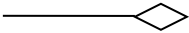
No	Simbol	Deskripsi	Keterangan
4		Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

c. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) [17].

Tabel II.3 Simbol *Class Diagram*

No	Simbol	Deskripsi	Keterangan			
1	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px;">nama kelas</td> </tr> <tr> <td style="padding: 2px;">+atribut</td> </tr> <tr> <td style="padding: 2px;">+operasi()</td> </tr> </table>	nama kelas	+atribut	+operasi()	<i>Class</i>	Kelas pada struktur sistem.
nama kelas						
+atribut						
+operasi()						

No	Simbol	Deskripsi	Keterangan
2		<i>Interface</i>	Sama dengan konsep interface dalam bahasa pemrograman berorientasi objek.
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4		<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5		<i>Generalisasi</i>	Relasi antar kelas dengan makna generalisasi atau spesialisasi.
6		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

II.5 Flowchart

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Rita Irviani, 2017).

Flowchart merupakan unit simbol gambar (*chart*) yang menunjukkan aliran (*flow*) dari proses terhadap data. *Flowchart* suatu gambar yang menjelaskan

urutan pembacaan data, pemrosesan data, pengambilan keputusan akhir dan penyajian hasil pemerosotan data.

Kesimpulan : *Flowchart* adalah Suatu Bagan yang menggambarkan arus logika dari data dalam suatu program dari awal sampai akhir. Bagan alir program merupakan alat yang berguna bagi programmer untuk mempersiapkan program yang rumit.

Tujuan membuat *flowchart* adalah :

1. Menggambarkan suatu tahapan penyelesaian masalah.
2. Secara sederhana, terurai, rapi dan jelas.
3. Menggunakan simbol-simbol standar.

Adapun jenis-jenis *flowchart* sebagai berikut [(Azwar, 2017):

1. *Flowchart* Sistem

Flowchart sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan didalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.

2. *Flowchart Paperwork*

Flowchart Paperwork sering disebut juga dengan *Flowchart* Dokumen, kegunaan utamanya adalah untuk menelusuri laur form laporan sistem dari satu bagian ke bagian lain baik bagaimana alur form dan laporan diproses, dicatat dan disimpan.

3. *Flowchart* Skematik

Flowchart skematik mirip dengan *flowchart* sistem yang menggambarkan suatu sistem prosedur. *Flowchart* Skematik ini bukan hanya menggunakan

simbolsymbol *flowchart* standar, tetapi juga menggunakan gambar-gambar komputer, form-form atau peralatan lain yang digunakan dalam sistem.



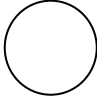

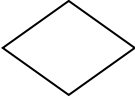
4. *Flowchart Program*

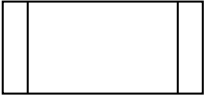
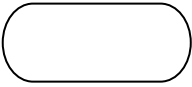

Flowchart Program merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur sesungguhnya dilaksanakan. *Flowchart* ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi.

5. *Flowchart Proses*

Flowchart proses merupakan teknik penggambaran rekayasa industrial yang memecah dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem.

Tabel II.4 Simbol-Simbol *Flowchart*

No	Simbol	Deskripsi	Keterangan
1		<i>Input/Output</i>	Merepresentasikan input data atau output data yang diproses atau informasi.
2		Proses	Mempresentasikan operasi.
3		Penghubung	Keluar ke atau masuk dari bagian lain <i>flowchart</i> khususnya halaman yang sama.
4		Anak Panah	Merepresentasikan alur kerja.
5		Keputusan	Keputusan dalam program

No	Simbol	Deskripsi	Keterangan
6		<i>Predefined Process</i>	Rincian operasi berada di tempat lain
7		<i>Terminal Points</i>	Awal atau akhir <i>flowchart</i>
8		Dokumen	Input/ouput dalam format yang dicetak

II.6 *Microsoft Visual Basic 2010*

Microsoft Visual Basic 2010 merupakan bagian dari kelompok bahasa pemrograman *Visual Studio 2010* yang dikembangkan oleh *microsoft*. *Visual studio 2010* terdiri dari beberapa bahasa pemograman diantaranya adalah *Microsoft Visual Basic 2010*, *Microsoft C# 2010*, *Microsoft Visual C++ 2010* dan *Visual Web Developer 2010*.

Selain itu kelebihan lain adalah memiliki *Object Relational Designer (O/R Designer)* untuk membantu mengedit LINQ ke SQL yang akan dihubungkan dengan database dan *fiture* lain, seperti WPF (*Windows Presentation Foundation*) dan WCF (*Windows Communication Foundation*). Semua hal yang baru tersebut akan menambah kelengkapan aplikasi *microsoft visual basic 2010* dalam membuat media dan dokumen.

Adapun komponen-komponen yang terdapat pada *microsoft visual basic 2010* yaitu: *label*, *textbox*, *button*, *listbox*, *checkboxlistbox* dan *checkbox*, disamping itu terdapat komponen *datetimepicker*, *monthcalender*, *radiobutton* dan *groupbox*.

Kumpulan komponen-komponen yang akan digunakan untuk membuat aplikasi disebut *toolbox*. Sedangkan tempat untuk meletakkan komponen-komponen yang terdapat pada *toolbox* disebut *form*.

Form juga sebagai tempat membuat program, *properties* adalah tempat untuk mengatur perintah atau komponen yang terdapat di dalam *form*. *Solution Explorer* adalah tempat meletakkan nama *project*, *Form* dan komponen lain yang berkaitan dengan aplikasi yang dibuat.

Pemrograman dengan *visual basic* dibuat dengan beberapa tahap berikut:

1. Menuliskan kode program dengan bantuan aplikasi IDE. Artinya, kita menuliskan program menggunakan aplikasi *Microsoft Visual Studio 2010* atau *Visual Basic 2010 Express Edition*.
2. Mengkompilasi kode program tersebut menjadi program yang dapat dijalankan.
3. Penyebaran (distribusi) program dikomputer dengan *NET Framework*.

II.7 *SQL Server 2008*

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. *SQL Server* adalah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. *Microsoft* merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment

pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu versi 32-bit (x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium IV atau lebih tepatnya prosesor 32 bit dan sistem operasi *Windows XP*).

- a. Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya *Core 2 Duo*) dan sistem operasi 64 bit seperti *Windows XP 64*, *Vista*, dan *Windows 7*. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini :
- b. Versi *compact*, ini adalah versi “Tipis” dari semua yang ada. Ini seperti versi *desktop* pada *SQL Sever 2000*. Versi ini juga digunakan pada *handled device* seperti *pocket PC*, *PDA*, *Smartphone*, *Tablet PC*.
- c. Versi *Express*, ini adalah versi “Ringan” dari semua yang ada (tetapi versi ini berbeda dengan versi *compact*) dan paling cocok untuk latihan. *Express Manager* standar, integrasi dengan CLR dan XML.

II.8 Crystal Report 8.5

Crystal Report merupakan program yang terpisah dengan program *Microsoft Visual Basic*, tetapi keduanya dapat dihubungkan (*Linkage*). Membuat laporan dengan *Crystal Report* hasilnya lebih baik dan lebih mudah, karena pada *Crystal Report* banyak tersedia objek-objek maupun komponen yang mudah digunakan.

Beberapa kelebihan dari *Crystal Report* ini adalah:

1. Dari segi pembuatan laporan tidak terlalu rumit yang memungkinkan para *programmer* pemula sekalipun dapat membuat laporan yang sederhana tanpa melibatkan banyak kode program.
2. Integrasi dengan bahasa-bahasa pemograman lain yang memungkinkan dapat digunakan oleh banyak programmer dengan masing-masing keahlian.
3. Fasilitas impor hasil laporan yang mendukung format-format populer seperti *Microsoft Word, Excel, Acces, Adobe Acrobat Reader, HTML*.