



BAB II

TINJAUAN PUSTAKA

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terkait

Penelitian terdahulu merupakan referensi bagi penelitian untuk melakukan penelitian ini.

1. Samuel Manurung, Vol. 9 No. 1 April 2018, ISSN ISSN: 2252-4983. Sistem Pendukung Keputusan Pemilihan Guru Dan Pegawai Terbaik Menggunakan MOORA. Hasil jurnal ialah sebuah sistem bagaimana memilih guru dan pegawai yang terbaik didalam sekolah dan untuk melakukan sebuah seleksi harus menggunakan dengan cara manual dan proses penilaian menjadi lama untuk mendapatkan hasil.
2. Haryanto, ISSN : 2442-7942 Vol. 4 Nomor 1 Tahun 2018. Pembuatan Aplikasi Sistem Penunjang Keputusan Untuk Pemilihan Penerima Beasiswa Siswa KMS Dengan Metode MOORA. Hasil jurnal ialah bahwa memberikan beasiswa kepada siswa pra-sejahtera dalam kajian ini digunakan berbagai macam kriteria yang telah ditentukan sebelumnya oleh pihak pemberi beasiswa dan pihak kampus dengan metode MOORA agar dapat menyeleksi siswa yang berhak mendapatkan beasiswa.
3. Mesran, dkk. Vol 2, No 2, April 2018 ISSN 2548-8368. Sistem Pendukung Keputusan Pemilihan Peserta Jaminan Kesehatan Masyarakat (Jamkesmas) Menerapkan Metode MOORA. Hasil jurnal ialah untuk menentukan yang

berhak menjadi peserta Jamkesmas berdasarkan kriteria dengan menggunakan rumus yang hasilnya lebih tepat sasaran dan lebih akurat.

4. Dwika Assrani, dkk. Vol. 5 No. 1, Februari 2018 ISSN 2407-389X. Penentuan Penerima Bantuan Siswa Miskin Menerapkan Metode Multi Objective Optimization on The Basis of Ratio Analysis (MOORA). Hasil jurnal ialah untuk membantu dalam pengambilan keputusan untuk siswa miskin yang berhak mendapatkan bantuan dengan lebih cepat dan lebih akurat.
5. Ramadiani, dkk. Vol. 6, No. 2, April 2019. Sistem Pendukung Keputusan Pemilihan Pramuka Pandega Berprestasi Menggunakan Metode Multi Objective Optimization On The Basis Of Ratio Analysis. Hasil jurnal ialah untuk memilih wakil pandega yang akan dikirim ke jakarta. Dengan pemilihan berdasarkan beberapa kriteria ditentukan oleh pengurus daerah.

II.2. Sistem Pendukung Keputusan

Kata Sistem berasal dari bahasa Yunani (*systema*) dan bahasa Latin (*systema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan. Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu (Syahrul Suci Romadhon dan Desmulyati, 2019).

Pengertian Sistem menurut Mulyadi adalah sekelompok dua atau lebih komponen-komponen yang saling berkaitan (subsistem-subsistem yang bersatu untuk mencapai tujuan yang sama).

Jadi pengertian sistem secara umum adalah jaringan kerja sama bagian-bagian atau unsur-unsur yang saling berhubungan guna mencapai tujuan yang diinginkan.

Definisi awal sistem pendukung keputusan adalah sebuah sistem untuk membantu seorang manajer dalam pengambilan keputusan dengan situasi semiterstruktur. Sistem pendukung keputusan dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas, namun tidak untuk menggantikan. (Ernawati, 2017)

Menurut Little, sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para seseorang (manajer, dokter, dll) dalam mengambil keputusan. Bahwa untuk sukses, sistem tersebut haruslah sederhana, cepat, mudah dikontrol, adaptif, lengkap dengan isu penting, dan mudah berkomunikasi. (Abdul Halim Hasugian dan Hendra Cipta, 2018)

Dengan pengertian diatas dapat dijelaskan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sistem pendukung keputusan ditujukan untuk membantu pihak manajemen dalam menganalisis situasi yang kurang terstruktur Sehingga sistem ini tidak

dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan.

II.2.1. Karakteristik Sistem Pendukung Keputusan

Ada beberapa karakteristik dari sistem pendukung keputusan, diantaranya adalah sebagai berikut (Rezqiwati Ishak, 2016) :

1. Dukungan untuk pengambil keputusan, terutama pada situasi semi terstruktur dan tak terstruktur.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk semua keputusan independen dan sekuensial.
5. Dukungan di semua fase proses pengambilan keputusan: intelegensi, desain, pilihan, dan implementasi.
6. Dukungan pada berbagai proses dan gaya pengambilan keputusan.
7. Kemampuan sistem beradaptasi dengan cepat dimana pengambil keputusan dapat menghadapi masalah-masalah baru dan pada saat yang sama dapat menanganinya dengan cara mengadaptasikan sistem terhadap kondisi-kondisi perubahan yang terjadi.
8. Pengguna merasa seperti di rumah. User-friendly, kapabilitas grafis yang kuat dan sebuah bahasa interaktif yang alami.
9. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, time lines, kualitas) dari pada efisiensi (biaya).

10. Pengambil keputusan mengontrol penuh semua langkah proses pengambilan keputusan dalam memecahkan masalah.
11. Pengguna akhir dapat mengembangkan dan memodifikasi situasi pengambilan keputusan.
12. Menggunakan model-model dalam penganalisisan situasi pengambilan keputusan.
13. Disediakkannya akses untuk berbagai sumber data, format dan tipe, mulai dari sistem informasi geografis (GIS) sampai sistem berorientasi objek.
14. Dapat dilakukan sebagai alat *stand alone* yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di satu organisasi keseluruhan dan di beberapa organisasi sepanjang rantai persediaan

II.2.2 Fungsi Sistem Pendukung Keputusan

Beberapa fungsi sistem pendukung keputusan sebagai berikut dibawah ini (Adi Widarma dan Hana Kumala, 2018) :

1. Sistem pendukung keputusan lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis, dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas.
2. Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatiskan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan dapat melakukan berbagai analisis dengan menggunakan model-model yang tersedia.

II.2.3 Langkah-Langkah Pengambilan Keputusan

Krismiaji menyatakan bahwa proses pembuatan keputusan melibatkan tahap-tahap berikut (Rico Darmanto, 2016) :

1. Mengidentifikasi dan mendefinisikan persoalan yang timbul.
2. Mengidentifikasi berbagai alternatif kemungkinan penyelesaian persoalan, dan mengeliminasi alternatif yang tidak layak atau tidak *feasible*.
3. Mengidentifikasi manfaat dan pengorbanan untuk setiap alternatif yang *feasible*. Klarifikasikan manfaat dan biaya tersebut ke dalam kelompok manfaat relevan dan manfaat tidak relevan, dan eliminasi manfaat yang tidak relevan.
4. Mengumpulkan data pendukung tentang seluruh biaya dan manfaat yang relevan, dan pastikan data tersebut terjadi pada periode atau rentang waktu yang sama.
5. Jumlahkan seluruh biaya relevan dan manfaat relevan untuk setiap alternatif.
6. Pilih alternatif terbaik, yaitu alternatif yang menghasilkan manfaat terbesar dan pengorbanan (biaya) terkecil (*least cost most benefit*).

II.3. Metode MOORA

Metode *Multi-Objective Optimization by Ratio Analysis* (MOORA) adalah metode yang diperkenalkan oleh Brauers dan Zavadkas. Metode yang relatif baru ini pertama kali digunakan oleh Brauers dalam suatu pengambilan dengan multi- kriteria. Metode MOORA memiliki tingkat fleksibilitas dan kemudahan untuk dipahami dalam memisahkan bagian subjektif dari suatu proses evaluasi kedalam kriteria bobot keputusan dengan beberapa atribut pengambilan keputusan. (Trismalia Hasanah, 2019)

Metode MOORA mudah dipahami dan fleksibel dalam memisahkan objek hingga proses evaluasi kriteria bobot keputusan. Metode MOORA juga memiliki tingkat selektifitas yang baik karena dapat menentukan tujuan dan kriteria yang bertentangan, yaitu kriteria yang bernilai menguntungkan (*Benefit*) atau yang tidak menguntungkan (*Cost*).

Berikut dibawah ini langkah penyelesaian metode MOORA yaitu sebagai berikut (Chairul Fadlan, 2019) :

1. Menginput nilai kriteria.
2. Membuat matriks keputusan
3. Normalisasi pada metode MOORA. Normalisasi bertujuan untuk menyatukan setiap element matriks sehingga element sehingga element pada matriks memiliki nilai yang seragam. Dengan menggunakan persamaan berikut ini:

$$X^{*ij} = \frac{X_{ij}}{\sqrt{[\sum_{i=1}^m X^2_{ij}]}}$$

Dimana X^{ij} merupakan nilai matriks keputusan x

4. Optimalkan Atribut. Untuk optimasi multi obyektif, pertunjukan normal ini ditambahkan dalam hal memaksimalkan (untuk menguntungkan atribut) dan dikurangi jika terjadi minimisasi (untuk atribut yang tidak menguntungkan).

Maka masalah optimasi menjadi:

$$Y_i = \sum_j^g = 1 X^* ij - \sum_j^n = g + 1 X^* ij$$

Dimana :

G merupakan atribut maksimum

X^{ij} merupakan nilai matriks keputusan x

Y^i merupakan nilai optimasi

5. Mengurangi nilai maximax dan minimax untuk menandakan bahwa sebuah atribut lebih penting itu bias dikalikan dengan bobot yang sesuai (koefisien signifikasi). Saat atribut bobot dipertimbangkan perhitungan menggunakan persamaan sebagai berikut.

$$Y_1 = \sum_j^g = W j X^* ij - \sum_j^n = g + 1 W j X^* ij$$

Dimana :

W merupakan nilai bobot kriteria

X^{ij} merupakan nilai matriks keputusan x

Y^i merupakan nilai optimasi

6. Menentukan ranking dari hasil perhitungan MOORA.

II.4. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan membuat *software* berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek. UML adalah salah satu *tool*/model untuk merancang pengembangan *software* yang berbasis *object oriented*.

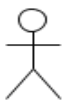



UML adalah keluarga notasi grafis yang didukung meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, Delphi dan bahkan dihubungkan secara langsung kedalam sebuah *object oriented database*.

II.4.1. Use Case Diagram

Use case atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

Tabel II.1. Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
2		<i>Association</i>	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
3		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
4		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).



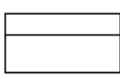

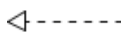


(Sumber : Irmayana, 2015)

II.4.2. *Class Diagram* (Diagram Kelas)

Class Diagram membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class Diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem.

Class Diagram memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram *Class* bersifat statis, menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. Simbol-simbol *Class Diagram* sebagai berikut dibawah ini :

Tabel II.2. Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya



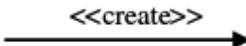
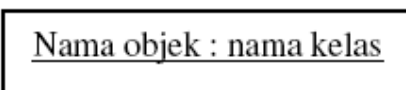

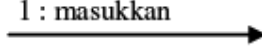
(Sumber : Aprianti, 2016)

II.4.3. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek.

Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel II.3 *Sequence Diagram*

No	Simbol	Deskripsi
1	Actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat
2	Garis Hidup/ <i>Lifeline</i> 	Menyatakan kehidupan suatu objek
3	Pesan Tipe <i>Create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
4	Objek 	Menyatakan objek yang berinteraksi pesan
5	Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi
6	Pesan Tipe <i>Send</i> 	Menyatakan bahwa suatu objek Mengirimkan data atau masukkan informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim






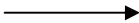
(Sumber : Rosa A.S dan M. Shalahudin, 2017)

II.4.4. *Activity Diagram*

Activity Diagram adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus, diagram aktivitas mempunyai peran seperti halnya diagram alur (*flowchart*), akan tetapi perbedaannya dengan *flowchart*, adalah diagram aktivitas bisa mendukung perilaku parallel sedangkan *flowchart* tidak bisa.

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan state diagram khusus.

Tabel II.4. Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Aktivitas menggambarkan proses yang berjalan, sementara <i>use case</i> menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas
2		<i>Start</i>	Mendefinisikan suatu tindakan sebelum aktivitas dimasukkan
3		<i>Activity Final Node</i>	Menandakan bahwa suatu tindakan atau aktivitas telah selesai
4		<i>Fork/Join</i>	Untuk mengilustrasikan proses-proses paralel (<i>fork</i> dan <i>join</i>) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal
5		<i>Decission</i>	Untuk menggambarkan <i>behaviour</i> pada kondisi tertentu.
6		<i>Control Flow</i>	Mendeskripsikan kemana aliran kegiatan berlangsung

(Sumber : Irmayana, 2015)

II.5. Basis Data (*Database*)

Basis data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada satu media, tanpa adanya suatu kerangkapan data, sehingga mudah untuk digunakan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, data disimpan sedemikian rupa sehingga apabila ada penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

Dari pengertian tersebut dapat disimpulkan bahwa Sistem Basis Data adalah Sistem yang terdiri atas kumpulan tabel/*file* yang saling berhubungan dalam sebuah basis data dan sekumpulan program berupa DBMS yang memungkinkan beberapa pemakai atau program lain untuk mengakses dan memanipulasi tabel-tabel tersebut.

Sistem basis data merupakan sistem penyusunan berkas data yang saling terpadu. Mempunyai komponen-komponen sebagai berikut:

1. *Database* (Basis Data)

Adalah kumpulan *file-file* yang saling berhubungan atau berelasi sehingga membentuk suatu basis data.

2. *Software* (Perangkat Lunak)

Adalah perangkat lunak yang digunakan dalam suatu sistem basis data. Pengelolaan basis data secara fisik tidak dapat dilakukan pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak yang khusus yang disebut DBMS (*Database Management System*) yang akan menentukan

bagaimana data diorganisasikan, disimpan, diubah dan diambil kembali.

Perangkat lunak yang termasuk dalam DBMS.

3. *Hardware* (Perangkat Keras)

Adalah perangkat keras dalam suatu sistem basis data, dimana mempunyai komponen-komponen utama berupa, Unit Pusat Pengolah (*Central Processing Unit* atau CPU), unit penyimpanan (*Storage Unit*), keyboard, monitor, printer dan lain-lain.

4. *Brainware* (manusia)

Manusia merupakan elemen penting pada sistem basis data. Pemakai ini terbagi atas empat kategori.

5. Administrator Basis Data

Yaitu tenaga ahli yang mempunyai tugas untuk mengawasi sistem basis data, merencanakan dan mengaturnya.

6. *Programmer*

Yaitu bertugas membuat program aplikasi yang diperlukan oleh pemakai akhir dengan menggunakan data yang terdapat dalam sistem basis data.

7. Pemakai Akhir

Yaitu tenaga ahli yang menggunakan data untuk mengambil suatu keputusan yang diperlukan dalam suatu instansi/perusahaan.

II.6. *Microsoft Visual Studio 2010*

Visual Basic merupakan bahasa pemrograman tingkat tinggi yang sudah sangat terkenal, dimulai dengan BASIC yang terdapat pada komputer “angkatan tua” seperti AT286. Pada saat itu bahasa BASIC merupakan bahasa yang sangat diandalkan dalam pembuatan beberapa aplikasi penting. BASIC digemari karena susunan pemrogramannya yang membebaskan kita untuk melompat dari satu baris ke baris lainnya.

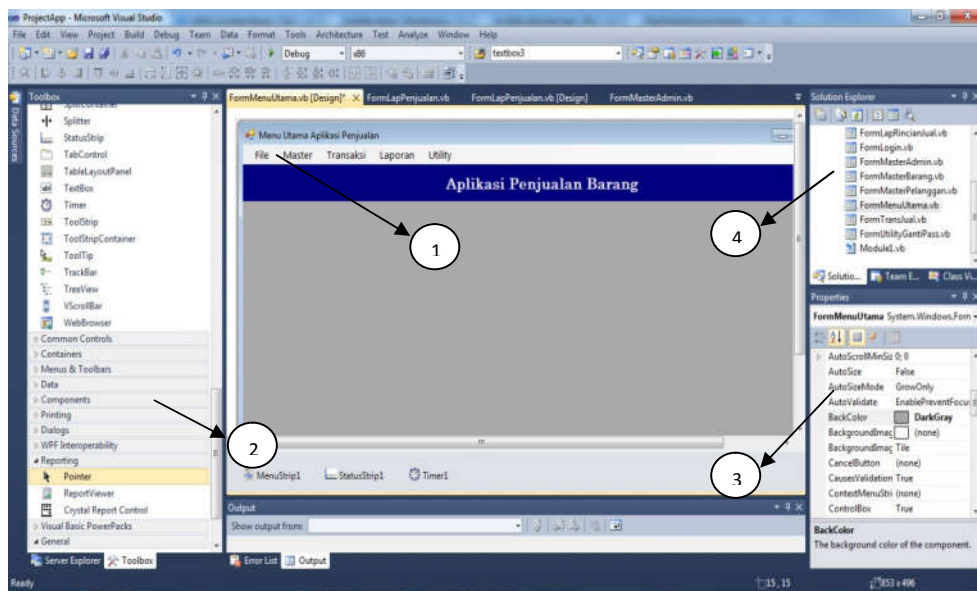
Dalam *Visual Studio 2008* terdapat beberapa *tool* yang dapat dipilih untuk pengembangan aplikasi. *Tool-tool* tersebut antara lain adalah *Visual Studio*, *Visual C#*, dan *Visual C++*. *Tool-tool* pada *Visual Studio 2008* tersebut menggunakan IDE (*Integrated Development Environment*) yang sama, sehingga dapat saling berbagi pakai fasilitas dalam pengembangan aplikasi.

Pada *Visual Studio 2010* banyak sekali fasilitas *wizard* yang disediakan untuk memudahkan para pengembang aplikasi. Dengan fasilitas ini pengembangan aplikasi dapat dilakukan dengan cepat. Ini memungkinkan para pemula untuk belajar lebih cepat dalam pengembangan aplikasi. Untuk mengembangkan aplikasi menggunakan *Visual Studio NET 2010* terlebih dahulu adalah mengenal IDE dari *Visual Studio NET 2010*.

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*.

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 *express edition*.
2. Mengkompilasi kode program tersebut menjadi program yang dapat dijalankan.
3. Penyebaran (distribusi) program dikomputer dengan *NET Framework*.



Gambar II.1. Tampilan Visual Basic 2010

(Sumber : javanetmedia.com)

Bagian-bagian pada jendela microsoft visual basic 2010 yaitu :

1. *Form*

Form adalah suatu objek yang dipakai sebagai tempat mengerjakan program aplikasi. *Form* ini berbentuk jendela dan dapat dibayangkan sebagai kertas atau meja kerja yang dapat dilukisi atau diletakkan ke dalam objek-objek lain.

2. *Toolbox*

Toolbox merupakan tempat menyimpan *control* yang akan digunakan untuk desain program. *Toolbox* ini terletak disebelah kiri *Visual Basic*.

3. *Properties*

Windows properties ini menyediakan segala properti dalam *project* yang diperlukan dalam perancangan *user interface*. Pada *windows* ini terdapat dua tab yang menampilkan properti dalam dua cara sesuai nama *tab*, yaitu *Alphabet* (diurutkan berdasarkan namanya sesuai abjad, ini merupakan pilihan *default*) dan *Category* (diurutkan berdasarkan fungsinya). Masing-masing properti memiliki nilainya sendiri-sendiri, yang telah disediakan *Visual Basic 2010*.

4. *Solution Explorer*

Solution explorer adalah keseluruhan desain yang dirancang dalam *form* atau dapat juga diartikan sebagai kumpulan dari *file* yang terorganisir dan membentuk sebuah program. *Project* disimpan dalam *file* yang berextension *vbp*. Jika kita membuat program aplikasi, akan terdapat jendela *project* yang berisi semua *file* yang dibutuhkan untuk menjalankan program *Visual Basic 2010*.

II.7. *Microsoft SQL Server 2008*

SQL Server adalah sistem manajemen database relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur client/server. Istilah client, server, dan client/server dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak. Pada level yang sangat umum, sebuah client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya. Sedangkan sebuah server adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai driver JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data *mirroring dan clustering*. Pada versi sebelumnya, MS SQL Server 2000 terserang oleh cacing komputer SQL Slammer yang mengakibatkan kelambatan akses Internet pada tanggal 25 Januari 2003.