

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian oleh Fatkhiana (2018), dengan judul "Pemanfaatan *Microsoft Access* Dalam Pembuatan Aplikasi Agenda Surat pada Divisi Pemasaran PT SIER". Dengan memanfaatkan Aplikasi Agenda Surat, diharapkan dapat membantu Divisi Pemasaran PT SIER dalam mengelola agenda surat, sehingga mengurangi waktu proses registrasi dan mempercepat Bagian Tata Administrasi dalam pengambilan keputusan. Perbedaan dalam penelitian tersebut dengan penelitian yang akan dilakukan hanya menggunakan *Microsoft access* dan tidak menggunakan *Visual Basic* sedangkan yang digunakan dalam penelitian ini adalah bahasa program *Visual Basic*.

Penelitian oleh Sari (2018), dengan judul "Sistem Informasi Pengarsipan Surat Masuk dan Surat Keluar di Notaris Debora Ekawati Lukman Dadali, SH." Sistem Informasi Pengarsipan Surat Masuk dan Surat Keluar berbasis komputer dapat mempermudah dalam proses pengolahan data surat masuk dan surat keluar, pengarsipannya dan pembuatan laporan. Perbedaan dalam penelitian tersebut dengan penelitian yang akan dilakukan terletak *tools* yang digunakan, yaitu menggunakan *PHP* dan *MYSQL* sedangkan *Tool & Methode* digunakan dalam penelitian ini adalah *Visual Basic*.

Penelitian oleh Prasetya (2017), dengan judul "Sistem Informasi Pengarsipan Data Surat Masuk dan Surat Keluar di Polrestabes Semarang". Dengan adanya fitur pembuatan laporan dan rekapitulasi dapat membantu mempercepat pembuatan laporan yang dibutuhkan oleh pimpinan, dan setiap laporan tersimpan secara terkomputerisasi dapat dicetak di saat dibutuhkan. Perbedaan dalam penelitian tersebut adalah sistem informasi berbasis *web* dan penelitian yang akan dilakukan berbasis *Graphical User Interface*.

2.2 Pengarsipan

2.2.1 Konsep Dasar Pengarsipan

Menurut Sayuti (2015) yang berbunyi :

- a. Naskah-naskah yang dibuat dan diterima oleh lembaga-lembaga negara dan badan-badan pemerintahan dalam bentuk corak apapun, baik dalam keadaan tunggal maupun berkelompok, dalam rangka pelaksanaan kegiatan pemerintahan,
- b. Naskah-naskah yang dibuat dan diterima oleh badan-badan swasta dan atau perorangan dalam bentuk corak apapun, baik keadaan tunggal ataupun berkelompok, dalam rangka pelaksanaan kehidupan kebangsaan. Kearsipan merupakan suatu kegiatan menempatkan dan menyimpan dokumen penting dalam suatu organisasi baik perusahaan maupun instansi yang bertujuan untuk menjamin keselamatan suatu organisasi sebagai barang bukti tertulis baik itu dalam bentuk gambar dan sebagainya.

2.2.2 Golongan Arsip

Menurut Sayuti (2015) arsip dapat dibedakan menjadi dua golongan yaitu arsip statis dan arsip dinamis.

a. Arsip dinamis

Arsip dinamis adalah arsip yang dipergunakan secara langsung dalam perencanaan, pelaksanaan, penyelenggaraan kehidupan kebangsaan pada umumnya atau dipergunakan secara langsung dalam penyelenggaraan Administrasi Negara. Jadi arsip dinamis adalah semua arsip yang berada di berbagai kantor, baik kantor pemerintah, swasta atau organisasi kemasyarakatan, karena masih dipergunakan secara langsung dalam perencanaan, pelaksanaan, dan kegiatan Administrasi lainnya. Arsip dinamis dalam bahasa Inggris disebut *record*.

b. Arsip statis

Arsip statis adalah arsip yang tidak dipergunakan secara langsung untuk perencanaan, penyelenggaraan kehidupan kebangsaan pada umumnya maupun untuk penyelenggaraan sehari-hari Administrasi Negara, dan kegiatan

Administrasi lainnya. Arsip statis adalah arsip yang disimpan di Arsip Nasional (ARNAS) yang berasal dari arsip dinamis dari berbagai kantor.

2.2.3 Tujuan Arsip

Menurut Sayuti (2015), mengatakan bahwa tujuan dari kearsipan adalah :

- a. Menyimpan surat dengan aman dan mudah selama diperlukan.
- b. Menyiapkan surat setiap kali saat diperlukan.
- c. Mengumpulkan bahan yang mempunyai sangkut paut dengan suatu masalah yang diperlukan sebagai pelengkap.

2.3 Pengertian Surat

2.3.1 Pengertian Surat Masuk

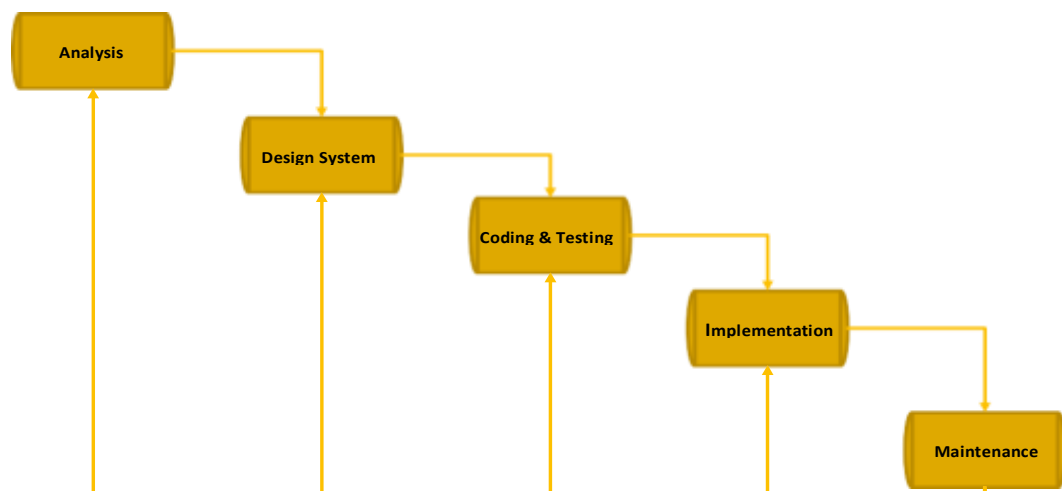
Menurut Barthos (2013) : “surat masuk adalah suatu alat komunikasi tertulis untuk menyampaikan pesan atau informasi dari satu pihak kepada pihak yang lain”. Surat masuk juga bisa didefinisikan sebagai semua tulisan dinas atau surat pribadi yang diterima instansi atau pihak lain untuk disampaikan pada pejabat yang tercantum pada alamat, baik yang tercantum pada sampul maupun tulisan dinas itu sendiri. Untuk memudahkan pengawasan maupun pengendaliannya, penerimaan surat masuk hendaknya dipusatkan di sekretariat atau bagian lain yang diberi wewenang melaksanakan tugas tersebut.

2.3.2 Pengertian Surat Keluar

Menurut Barthos (2013), “surat keluar adalah surat yang lengkap (bertanggal, bernomor, berstempel, dan ditandatangani oleh pejabat yang berwenang) yang dibuat oleh suatu instansi atau lembaga lain”. Surat keluar biasanya dikirim melalui pos atau kurir pada surat biasanya berisikan berupa pemberitahuan, undangan pertemuan atau rapat, undangan kerjasama ataupun jawaban atau konfirmasi dari surat yang telah dikirim sebelumnya. Pada surat itu juga ada yang berisikan surat perintah tugas.

2.4 System Development Life Cycle (SDLC)

Menurut Pressman (2012) : *System Development Life Cycle* adalah suatu pendekatan yang sistematis dan berurutan. Suatu proses pengembangan perangkat lunak yang sistematis dan berurutan, suatu kemajuan dipandang sebagai terus mengalir ke bawah melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian disebut dengan metode *Waterfall*. Berikut ini adalah gambaran pengembangan perangkat lunak metode *Waterfall*.



Gambar 2.1 System Development Life Cycle

Berikut adalah penjelasan tahapan metode *Waterfall* :

1. *Analysis*

Merupakan langkah analisis terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini melakukan sebuah penelitian, wawancara atau *study literatur*. Tahapan ini akan menghasilkan *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. *User requirement* ini yang akan menjadi acuan sistem analisis untuk diterjemahkan ke dalam bahasa pemrograman.

2. *Design System*

Desain system merupakan tahapan untuk merancang desain *database*, *user interface*, dan *report* dari proyek sistem informasi yang akan dibangun. Tahapan ini merupakan tahapan yang penting bagi pembuatan proyek sistem informasi, karena *output* yang akan dihasilkan pada tahap ini menentukan berjalan atau tidaknya sistem yang akan dibuat.

3. *Coding dan Testing*

Coding merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat.

Testing ini digunakan untuk menemukan kesalahan-kesalahan terhadap sistem dan kemudian bisa diperbaiki. Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem.

4. *Implementation*

Implementation merupakan tahapan untuk menerapkan sistem yang telah dibangun guna menggantikan proses bisnis yang ada saat ini.

5. *Maintenance*

Tahap pemeliharaan merupakan tahapan dimana akan dilakukan perbaikan serta manajemen kembali sistem informasi yang telah diterapkan yang bertujuan guna pengembangan sistem. Tahapan ini juga digunakan untuk menjaga kinerja sistem yang telah diterapkan agar dapat berjalan sesuai dengan optimal.

2.5 *Data Flow Diagram*

“*Data Flow Diagram* atau *DFD* adalah alat yang menunjukkan alur data pada sistem dalam bentuk grafik. Elemen penting dari *DFD* adalah alur data, proses, penyimpanan data dan sumber data. Sistem analisis membuat *DFD* berdasarkan *level*. *DFD level* tinggi hanya mengidentifikasi proses besar”. (Irwansyah, 2014:240).

Sedangkan menurut Sukanto dan Shalahudin (2013: 69) menjelaskan bahwa “*Data Flow Diagram* adalah representasi grafik yang menggambarkan aliran informasi dan transformasi yang diaplikasikan sebagai data yang mengalir dari masukan dan keluaran”.

Dari definisi di atas dapat disimpulkan *Data Flow Diagram* merupakan alat yang digunakan untuk menggambarkan suatu sistem, selain itu *DFD*

merupakan dokumentasi dari sistem yang baik karena menggambarkan arus data secara jelas. *DFD* ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisis maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Al-Bahra bin Ladjamuddin (2005: 67-75), elemen dasar dari *DFD* adalah sebagai berikut :

1. Kesatuan Luar (*External Entity*)

Kesatuan luar adalah sesuatu yang berada di luar sistem tetapi memberikan data ke dalam sistem atau sebaliknya. Kesatuan luar tidak termasuk bagian dari sistem. Pedoman pemberian nama kesatuan luar adalah sebagai berikut:

- a. Nama kesatuan luar berupa kata benda.
- b. Kesatuan luar tidak boleh memiliki nama yang sama kecuali memang objeknya sama.

2. Proses (*Process*)

Proses merupakan kegiatan atau kerja yang dilakukan oleh sistem. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Pedoman pemberian nama proses adalah sebagai berikut:

- a. Nama proses terdiri dari kata kerja dan kata benda yang mencerminkan fungsi proses tersebut.
- b. Kata proses tidak boleh digunakan sebagai bagian dari nama suatu proses.
- c. Tidak boleh ada beberapa proses yang memiliki nama yang sama.
- d. Proses harus diberi nomor.

3. Simpanan Data (*Data Store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Pedoman pemberian nama simpanan data adalah sebagai berikut :

- a. Nama harus mencerminkan simpanan data tersebut.
- b. Bila namanya lebih dari satu kata, maka harus diberi tanda sambung.

4. Arus Data (*Data Flow*)


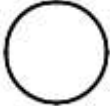


Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Pedoman pemberian nama aliran data adalah sebagai berikut:

- Nama aliran data yang terdiri dari beberapa kata dihubungkan dengan garis sambung.
- Sedapat mungkin nama aliran data ditulis lengkap.
- Tidak boleh ada aliran data dari kesatuan luar ke simpanan data atau sebaliknya, hubungan kesatuan luar dengan simpanan data harus melalui proses.

5. Simbol *DFD*

Simbol *DFD* disajikan pada Tabel 2.1

Tabel 2.1. Simbol Data Flow Diagram

Simbol	Keterangan
	Kesatuan Luar
	Proses
	Arus Data
	Penyimpanan Data

2.5.1 Diagram Konteks

Menurut Jogiyanto, (2010:699) dalam buku Analisa Sistem Informasi, menjelaskan bahwa : “Diagram Konteks adalah diagram yang menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut digunakan untuk menggambarkan sistem secara umum atau global dari keseluruhan sistem yang ada”.







2.5.2 Flowmap

Menurut Winarno (2004:102) *flowmap* adalah gambar yang menggunakan lambang-lambang baku untuk menggambarkan sistem atau proses. Pada waktu akan menggambar suatu bagan alir, analisis sistem dapat mengikuti pedoman sebagai berikut :

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di bagan alir harus ditunjukkan dengan jelas darimana kegiatan akan dimulai dan dimana akan berakhir.
3. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
4. Masing-masing kegiatan di dalam bagan alir harus didalam urutan yang semestinya.
5. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas.
6. Gunakanlah simbol-simbol bagan alir yang standar.

Simbol-simbol *flowmap* ditunjukkan pada Tabel 2.2 berikut :

Tabel 2.2 Simbol Flowmap

Simbol	Keterangan	Simbol	Keterangan
	Mulai/selesai		Kondisi
	Proses		Tempat Penyimpanan
	Dokumen input/output		Penghubung

2.5.3 Entity Relationship Diagram (ERD)

Salah satu *tools diagram* yang digunakan untuk memodelkan abstraksi data adalah *Entity Relationship Diagram (ERD)*.




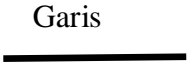
Menurut Mulyani (2016:100) mengungkapkan :

EntityRelationship Diagram (ERD) adalah *tools* yang digunakan untuk melakukan pemodelan data secara abstrak dengan tujuan untuk mendeskripsikan atau

menggambarkan struktur dari data yang digunakan. Adapun fungsi utama *ERD* yaitu sebagai alat untuk memodelkan hasil dari analisis data, sebagai alat untuk memodelkan data konseptual dan sebagai alat untuk memodelkan objek dalam suatu sistem.

Adapun simbol-simbol ERD dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-Simbol ERD

Notasi	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah)
	Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut.

2.6 Database

Menurut Wasito (2010), mengungkapkan *Database* dapat diterjemahkan sebagai basis data, sehingga menjadi dua kata yaitu Basis dan Data. Basis adalah tempat adanya sekumpulan. Data adalah representasi fakta dari dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa yang direkam dalam bentuk angka, huruf, teks, gambar atau kombinasinya. Secara sederhana *database* didefinisikan sebagai kumpulan informasi terintegrasi, diorganisasikan dan disimpan dalam suatu cara yang memudahkan pengambilan kembali. Sekumpulan informasi tersebut disusun secara teratur berdasarkan suatu syarat atau subyek tertentu.

2.6.1 Objek Database

Menurut Agency (2015), mengungkapkan *Database* menyatakan suatu wadah untuk mengelola data. *Database* mencakup sejumlah tabel dan berbagai objek yang terkait dengan pengelolaan data. Objek selain tabel dapat berupa formulir (*form*), *query*, laporan (*report*), halaman (*page*), makro dan modul. Penjelasan mengenai objek *database* seperti dilihat pada Tabel 2.4.

Tabel 2.4 Objek Database

Objek Database		Keterangan
1.	<i>Table</i> (Tabel)	adalah objek yang digunakan untuk menyimpan kumpulan data yang terkait.
2.	<i>Query</i> (Queri)	adalah suatu bentuk untuk memperoleh informasi dari <i>database</i> .
3.	<i>Form</i> (Formulir)	adalah tampilan atau antar muka yang memudahkan dalam memasukkan data atau menyajikan informasi.
4.	<i>Report</i> (Laporan)	adalah dokumen yang digunakan untuk menyajikan informasi atau <i>database</i>
5.	<i>Pages</i> (Halaman)	adalah objek yang memungkinkan pengaksesan informasi dari <i>database</i> ke dalam bentuk halaman web.
6.	<i>Macros</i>	adalah berisi kumpulan kode yang digunakan untuk mengotomasikan suatu tindakan tertentu dengan sekali langkah.
7.	<i>Modules</i>	adalah program yang ditulis dengan menggunakan VBA (<i>Visual Basic for Applications</i>) yang ditunjukkan untuk memperluas sebuah fungsi <i>database</i> .

2.6.2 Ketentuan Tipe Data

Tiap *field* pada tabel harus didefinisikan tipe datanya. Penentuan tipe data dilakukan pada saat desain tabel. seperti dilihat pada Tabel 2.5.

Tabel 2.5 Tipe Data Field

Data Field		Keterangan
1.	<i>Text</i>	Jenis data yang terdiri dari sejumlah karakter. karakter dapat terdiri huruf, angka, tanda baca dll. Panjang maksimum adalah 255 karakter.
2.	<i>Number</i>	Jenis data yang berisi khusus angka yang digunakan untuk operasi logika dan matematika. Misalnya untuk penjumlahan, pengurangan, ekspresi logika.
3.	<i>Date/Time</i>	Jenis data yang berisi khusus tanggal dan waktu. jenis data ini dapat digunakan untuk operasi yang terkait dengan matematika tanggal.
4.	<i>Currency</i>	Jenis data yang berisi khusus angka dalam format mata uang jenis data dapat digunakan untuk operasi yang terkait matematika mata uang.
5.	<i>Yes/No</i>	jenis data yang berisi khusus angka yang diformat secara urut saat mengisi <i>record</i> .
6.	<i>Autonumber</i>	Jenis data yang terdiri dari dua pilihan.
7.	<i>Hyperlink</i>	Jenis data untuk koneksi berupa <i>e-mail</i> , <i>website</i>
8.	<i>Lookup Wizard</i>	digunakan untuk menampilkan pilihan pengisian record dalam bentuk <i>listbox</i> atau <i>combo box</i> .

2.6.3 Model Relasional

Menurut Wasito (2010), menyatakan Konsep *database* dalam suatu rancangan tidak lepas dari model relasional. Unit dasar organisasi pada sejumlah *database* relasional adalah tabel. Tabel adalah pusat *database*. Kolom yang mewakili *field-field* dan baris mewakili *record*. *Access* menyimpan setiap *entry database* pada barisnya sendiri, yang disebut dengan *record*.

Record adalah sekumpulan informasi tentang suatu hal. Sejumlah tabel dapat terhubung satu sama lain berdasarkan *field* yang sejenis. Koneksi antar tabel tersebut merupakan sebuah metode yang disebut dengan *relationship*. *Relationship* memiliki tiga bentuk yaitu :

1. *One-to-one Relationship*

Hubungan antara tabel pertama dengan tabel kedua adalah satu berbanding satu. dalam hal ini, tiap *record* dalam tabel pertama hanya memiliki satu *record* yang cocok pada tabel kedua. Hubungan satu ke satu ini digunakan untuk membuat satu tabel dengan banyak *field*, untuk memisahkan sebagian tabel dalam mengamankan data dan untuk menghemat penyimpanan data yang hanya dioperasikan pada sebagian isi tabel.

2. *One-to-many Relationship*

Hubungan antara tabel pertama dengan tabel kedua adalah satu berbanding banyak. Dalam hal ini, tiap *record* dalam tabel pertama memiliki lebih dari satu *record*.

3. *Many-to-many Relationship*

Hubungan antara tabel pertama dengan tabel kedua adalah banyak berbanding banyak. dalam hal ini, sejumlah *record* dalam tabel pertama juga memiliki lebih dari satu *record* yang cocok pada tabel kedua.

2.7 Microsoft Access dalam Pengelolaan Surat

Menurut Andi (2012), menjelaskan “*Microsoft Access* merupakan program aplikasi *database* yang mudah digunakan baik untuk menyortir, mengatur data, membuat tabel data serta melaporkan informasi penting. *Microsoft Access* adalah *software* yang dapat membuat dan mengolah data dari menyortir, mengatur, serta membuat tabel data yang menampilkan hasil akhir berupa laporan. Seperti *database* yang berbasis *relational*, *Access* memberi kemudahan dalam melakukan koneksi antar informasi yang tersimpan. *Access* juga mampu bekerjasama dengan program *database* yang lebih dulu populer seperti *dBase* dan *Paradox*.

Menurut Wasito (2010), menjelaskan Di lingkungan sistem operasi *Microsoft Office*, *Access* memiliki desain dan pengembangan aplikasi olah data tanpa perlu mengetahui dasar pemrograman kecuali konsep *database*. *Database* dalam *Access* adalah sekumpulan data atau informasi yang disimpan dalam sebuah tabel. Tabel yang memiliki satu keterkaitan cukup disimpan dalam satu file *database*.

2.8 Visual Basic Net 2010

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu.

Visual Studio 2010 (yang sering juga disebut dengan *VB .Net 2010*) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis windows. Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol *ActiveX*, *file Help*, aplikasi Internet dan sebagainya.
3. Menguji program (debugging) dan menghasilkan program berakhiran EXE yang bersifat *executable* atau dapat langsung dijalankan.

Visual Studio 2010 adalah bahasa yang cukup mudah untuk dipelajari. Bagi programmer pemula yang baru ingin belajar program, lingkungan Visual Studio dapat membantu membuat program dalam sekejap mata. Sedang bagi programmer tingkat lanjut, kemampuan yang besar dapat digunakan untuk membuat program-program yang kompleks, misalnya lingkungan net-working atau client server.

Bahasa Visual Studio cukup sederhana dan menggunakan kata-kata bahasa Inggris yang umum digunakan. Kita tidak perlu lagi menghafalkan sintaks-sintaks maupun format-format bahasa yang bermacam-macam, di dalam Visual Basic semuanya sudah disediakan dalam pilihan-pilihan yang tinggal diambil sesuai dengan kebutuhan. Selain itu, sarana pengembangannya yang bersifat visual memudahkan kita untuk mengembangkan aplikasi berbasis Windows, bersifat *mouse-driven* (digerakkan dengan *mouse*) dan berdaya guna tinggi.

2.8.1 Sejarah Singkat Visual Studio 2010

Berikut ini beberapa point penting dalam sejarah perkembangannya :

1. Pertama kali di release dengan nama Visual Basic yang dikeluarkan pada tahun 1991, yaitu Visual Basic yang masih berbasis DOS dan untuk Windows
2. Visual Basic 3.0 dirilis pada tahun 1993
3. Visual Basic 4.0 dirilis pada tahun 1994 dengan dukungan untuk aplikasi 32 bit
4. Visual basic 6.0 dirilis pada akhir tahun 1998
5. Visual basic untuk selanjutnya yaitu versi .Net yang dirilis awal 2002
6. Pada tahun 2003 Visual Studio 2003 dirilis untuk memperbaiki kinerja dari visual Studio 2002 dengan meluncurkan .NET Framework versi 1.1.
7. Pada tahun 2005 Microsoft mengeluarkan Visual Basic Versi 8.0 atau biasa di sebut Microsoft Visual Studio 2005.
8. Pada tahun 2008 Microsoft juga mengeluarkan versi 9.0 atau biasa di sebut Microsoft Visual Studio 2008.
9. Hingga pada tahun 2010 Visual Studio sudah pada versi 10.0 atau biasa di sebut Microsoft Visual Studio 2010, dimana didalamnya sudah ditambahkan bahasa pemrograman yang baru yaitu F# yang merupakan penyempurnaan dari versi-versi sebelumnya.

2.8.2 Keistimewaan Visual Studio 2010

Beberapa keistimewaan Visual Studio 2010 ini diantaranya seperti :

1. Menggunakan platform pembuatan program yang dinamakan developer studio, yang memiliki tampilan dan sarana yang sama dengan Visual C++ dan Visual J++. Dengan begitu Anda dapat bermigrasi atau belajar bahasa pemrograman lainnya dengan mudah dan cepat.
2. Memiliki compiler handal yang dapat menghasilkan file executable yang lebih cepat dan lebih efisien dari yang sebelumnya.
3. Memiliki beberapa tambahan wizard yang baru. Wizard adalah sarana yang mempermudah di dalam pembuatan aplikasi dengan mengotomisasi tugas-tugas tertentu.

4. Visual Studio 2010 mempunyai beberapa fitur untuk pengembangan berbagai macam aplikasi yang diantaranya; Windows Development, Web Development, Office Development, Sharepoint Development, Cloud Development (Windows Azure), Silverlight Tooling, Multi-Core Development, Customizable IDE.

Sumber : <https://hazistrihartoyo.wordpress.com/2015/11/19/pengertian-dan-sejarah-microsoft-visual-studio-2010/>

2.9 Black Box Testing

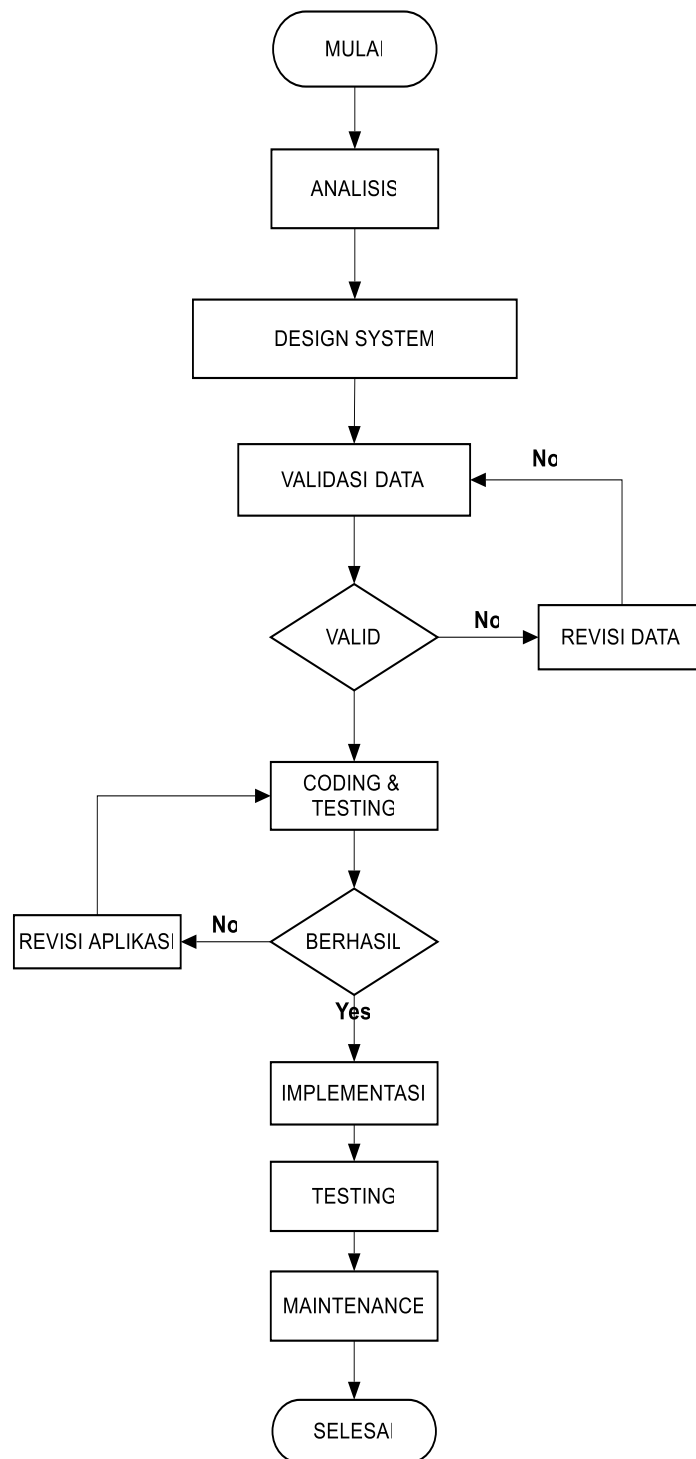
Menurut Pressman (2010:597), menjelaskan : *Black Box Testing* (Pengujian Kotak Hitam), juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian kotak hitam memungkinkan anda untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam bukan teknik alternatif untuk kotak hitam. Sebaliknya, ini merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari yang diungkap oleh metode kotak putih.

Menurut Pressman (2010:597), Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang salah atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses basis data eksternal.
4. Kesalahan perilaku atau kinerja
5. Kesalahan inisialisasi dan penghentian

2.10 Kerangka Pemikiran

Melihat dari kelemahan sistem manual dalam proses kegiatan pengarsipan surat masuk dan surat keluar pada SMK Muhammadiyah 1 Kedungtuban, dibutuhkan suatu program terstruktur guna menunjang kegiatan operasional program pengarsipan surat masuk dan surat keluar melalui tahapan dalam proses perancangan program. Implementasi dapat dilihat pada Gambar 2.2.



Gambar 2.3 Kerangka Pemikiran

Keterangan

1. *Analysis*

Tahapan analisis dilakukan dengan mengumpulkan data melalui wawancara atau *study literatur* mengenai sistem yang berjalan. Tahapan ini menghasilkan *user requirement* untuk mengetahui kebutuhan sistem yang diusulkan.

2. *Design System*

Desain sistem merupakan tahapan untuk merancang desain *database*, *user interface*, dan *report* dari aplikasi yang akan dibangun. Aplikasi ini menggunakan basis data Microsoft Access 2007 format accdb dengan membuat interface menggunakan Microsoft Visual Studio 2010 (VB Net)

3. *Coding dan Testing*

Dalam tahapan ini dilakukan pengkodean terhadap form input data dan report yang dibutuhkan dalam aplikasi ini.

Testing ini digunakan untuk menemukan kesalahan-kesalahan terhadap sistem dan kemudian bisa diperbaiki. Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem.

4. *Implementation*

Implementation merupakan tahapan untuk menerapkan aplikasi yang telah dibangun yaitu dengan melakukan input data dari form yang telah dibuat.

5. *Maintenance*

Tahap pemeliharaan merupakan tahapan dimana akan dilakukan perbaikan serta manajemen kembali sistem informasi yang telah diterapkan yang bertujuan guna pengembangan sistem. Tahapan ini juga digunakan untuk menjaga kinerja sistem yang telah diterapkan agar dapat berjalan sesuai dengan optimal.

