

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya dengan judul “Sistem Pembayaran SPP di SD Muhammadiyah 16 Surakarta Berbasis Aplikasi Desktop” yang dilakukan oleh Ray Septy Fajar Muhammad. Aplikasi pembayaran SPP ini dibuat menggunakan *software Visual Basic 6 (VB6)* dan *software crystal report* yang menggunakan metode *System Development Life Cycle (SDLC)* model *waterfall*. Pembuatan aplikasi ini bertujuan untuk mempermudah guru bagian keuangan agar pekerjaannya lebih efektif dan efisien serta mudah dalam proses pelayanan pembayaran. [1]

Penelitian lain yang berjudul “Perancangan Aplikasi Administrasi Sekolah Berbasis Desktop” oleh Ahmad Musyafa. Sistem ini dirancang untuk memberikan laporan keuangan siswa dan memberikan informasi keuangan secara *up to date*. Sistem Informasi ini dikembangkan dengan menggunakan bahasa pemrograman *PHP* dan menggunakan *MYSQL* sebagai basis data. [2]

Penelitian lain berjudul “Perancangan Sistem Informasi Pembayaran SPP pada SMK NEGERI 04 Pekanbaru Berbasis Desktop” oleh Yunahar Heriyanto dan Nelly Oktavianis. Sistem ini dibangun untuk memudahkan staff tata usaha dalam proses pengolahan data pembayaran serta pencarian data SPP dan memudahkan dalam pembuatan laporan keuangan dengan cepat dan akurat. Sistem Informasi ini dikembangkan dengan menggunakan *software Visual Basic 6.0* dan menggunakan *MYSQL* sebagai basis data.[3]

Penelitian yang akan dilaksanakan oleh penulis berbeda dengan penelitian sebelumnya. Pada penelitian ini penulis bermaksud untuk membuat Sistem Informasi Administrasi Pembayaran Biaya Sekolah Berbasis Desktop untuk membantu pegawai tata usaha dalam mengelola pembayaran biaya sekolah serta memudahkan kepala sekolah dalam melihat laporan keuangan mengenai pembayaran sekolah, serta menambahkan fungsi *SMS Gateway* untuk memberikan notifikasi kepada wali murid. Metode yang digunakan untuk membuat aplikasi ini adalah metode *prototype*.Sistem ini dibuat menggunakan *software* Microsoft Visual Studio 2010 dengan bahasa pemrograman *C#*.

Penelitian ini diperlukan adanya teori-teori yang mendasar untuk menunjang proses penelitian ini. Teori-teori yang digunakan dalam penelitian ini adalah :

2.2 Landasan Teori

2.2.1 Sistem

Sistem dapat diartikan sebagai satu kesatuan yang terdiri dari komponen-komponen atau subsistem yang tertata dengan teratur, saling interaksi, saling ketergantungan satu dengan yang lainnya, dan tidak dapat dipisahkan (integratif) untuk mewujudkan suatu tujuan. Hal ini pun dikemukakan oleh Tata Sutabri bahwa sistem dapat didefinisikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Maksud dari komponen atau subsistem di dalam pengertian ini tidak hanya komponen fisik semata, tetapi termasuk di dalamnya adalah komponen yang bersifat abstrak atau komponen secara konseptual, seperti visi, misi, kebijakan, prosedur, dan kegiatan informal lainnya. [4]

Sesuatu dikatakan sebagai suatu sistem apabila memiliki sifat-sifat tertentu seperti dikemukakan oleh Jogiyanto, sistem memiliki karakteristik atau sifat-sifat tertentu, yakni berikut ini.

- a. Mempunyai komponen-komponen (*components*)
Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen dapat berupa suatu subsistem atau bagian-bagian dari sistem.
- b. Batas sistem (*boundary*)
Setiap sistem memiliki batas-batas luar yang memisahkannya dari lingkungannya. Batas sistem adalah wilayah yang membatasi antara satu sistem dengan sistem yang lainnya atau dengan lingkungannya. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.
- c. Lingkungan luar sistem (*enviromments*)
Lingkungan luar adalah lingkungan di luar batas sistem yang mempengaruhi operasi sistem. Pengaruh tersebut dapat bersifat positif atau negatif suatu sistem tersebut. Pengaruh yang positif dapat dipelihara dan dijaga, sedangkan pengaruh negatif harus dikendalikan karena dapat mengganggu sistem.
- d. Penghubung sistem (*interface*)

- Penghubung adalah media yang menghubungkan atau mengintegrasikan antara satu subsistem ke subsistem yang lainnya menjadi satu kesatuan.
- e. Masukan sistem (*input*)
Masukan adalah serangkaian data (*signal input*) atau maintenance input dari dalam atau dari luar lingkungan untuk diolah dalam sistem untuk dioperasikan. Contoh di dalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.
 - f. Keluaran sistem (*output*) Keluaran adalah hasil dari proses dan diklasifikasi menjadi keluaran yang berguna. Keluaran merupakan masukan untuk subsistem yang lain. Informasi adalah keluaran yang dihasilkan dari proses.
 - g. Pengolah sistem (pemrosesan)
Pengolah merupakan suatu yang merubah masukan menjadi keluaran. Contoh Sistem akuntansi akan mengolah data-data transaksi menjadi laporan keuangan yang diperlukan oleh manajemen.
 - h. Sasaran sistem
Sistem yang baik tentu memiliki sasaran yang ingin dicapai. Sasaran adalah sesuatu yang menjadi target yang ingin dicapai dari suatu sistem. Sasaran yang dicapai dari suatu sistem menentukan masukan yang dibutuhkan. Suatu sistem dikatakan berhasil apabila sasaran yang telah ditentukan dapat dicapai dengan baik.

2.2.2 Informasi

Informasi adalah sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.[4]

2.2.3 Sistem Informasi

Sistem Informasi dapat didefinisikan sebagai berikut [4]:

1. Suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

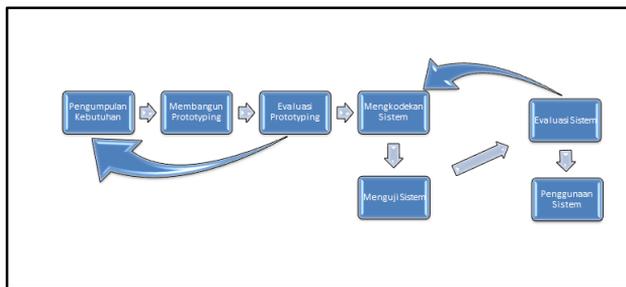
2. Satuan komponen yang saling berhubungan yang mengumpulkan (atau mendapatkan kembali), memproses, menyimpan, serta mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi.

2.2.4 Laporan Keuangan

Laporan keuangan pada dasarnya merupakan hasil proses akuntansi yang dapat digunakan sebagai alat untuk berkomunikasi antara data keuangan dengan pihak-pihak yang berkepentingan, sehingga laporan keuangan memegang peranan yang luas dan mempunyai suatu posisi yang mempengaruhi dalam pengambilan keputusan. [5]

2.2.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang bekerja secara efisien menggunakan mesin. Perangkat lunak sering dibuat dan pada akhirnya tidak memenuhi kebutuhan pelanggan atau bahkan karena masalah non-teknis seperti keengganan pemakai perangkat lunak (*user*) untuk mengubah kerja manual ke otomatis.



Gambar 2. 1 Metode Prototyping

Metode Prototype merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau waterfall development model. [6]

Berikut adalah tahap-tahap dalam metode *prototyping*

1. **Analisa kebutuhan**
Pada tahap ini analisa kebutuhan akan dilakukan secara langsung ke tempat penelitian yaitu SMP Al-Manaar CAkan dilakukan pengumpulan data yang menjadi kebutuhan dasar dari sistem yang akan dibuat.
2. **Membangun *prototyping***
Pada tahap ini akan dibangun sebuah *prototype* berupa tampilan *input* dan *output* sesuai dengan data yang sudah didapat pada analisa kebuthan.
3. **Evaluasi *prototyping***
Prototype kemudian di evaluasi pelanggan untuk mengetahui apakah *prototype* yang dibangun sudah sesuai dengan keinginan pelanggan. Apabila *prototype* dianggap belum sesuai maka akan dilakukan perbaikan model *prototype* dan kemudian akan dilakukan evaluasi kembali.
4. **Mengkodekan sistem**
Prototype yang sudah disetujui selanjutnya akan dibuat menjadi sistem dengan bahasa pemrograman C#.
5. **Menguji sistem**
Sistem yang sudah dibuat kemudian akan diuji dengan menggunakan metode *black box*.
6. **Evaluasi sistem**
Sistem yang sudah diuji menggunakan metode *black box* selanjutnya akan dievaluasi oleh pelanggan apakah sistem yang sudah jadi sesuai dengan yang diharapkan. Jika tidak sesuai maka akan kembali ke tahap pengkodean sistem dan pengujian sistem.
7. **Menggunakan sistem**
Sistem yang sudah dievaluasi dan diuji oleh pelanggan siap untuk digunakan.

Metode pengujian yang digunakan dalam penelitian ini adalah *Black Box Testing*. Pengujian kotak hitam (*black box testing*), juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. 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, dan (5) kesalahan inisialisasi dan penghentian.

Flowchart adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah dipahami, mudah digunakan dan standar. Tujuan penggunaan flowchart adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai dan rapi dengan menggunakan symbol-simbol yang mudah dipahami oleh programmer. Berikut adalah simbol-simbol pada *flowchart* seperti pada Tabel 2.1.[7]

Tabel 2. 1 Simbol - simbol *flowchart*

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program.
2.		<i>Input – Output</i>	Memasukan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya.
3.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
4.		<i>Flow</i>	Menghubungkan antara symbol satu dengan symbol yang lain atau menyatakan jalannya arus dalam suatu proses. symbol arus ini sering

			disebut dengan <i>connecting line</i> .
5.		<i>Document</i>	Merupakan symbol untuk data yang terbentuk informasi.
6.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
7.		<i>Manual Input</i>	Memasukan data secara manual <i>on-line keyboard</i> .

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek. [8]

1. *Use Case Diagram*

Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem ini berinteraksi dengan dunia luar, misalnya menyusun sebuah daftar layanan kesehatan. *Use Case Diagram* dapat digunakan untuk memperoleh kebutuhan sistem dan memahami bagaimana sistem seharusnya bekerja. Simbol-simbol *Use Case Diagram* bisa dilihat pada Tabel 2.2. [8]

Tabel 2. 2 Simbol-Simbol Use Case Diagram

No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Deskripsi dari urutan aksi –aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

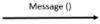
2.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas

2. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek berupa pesan (message) yang digambarkan terhadap waktu. Sequence diagram terdiri antar dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait). Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya

eksekusi sebuah proses. Berikut adalah simbol-simbol yang ada pada *sequence diagram*, sesuai pada Tabel 2.3.[8]

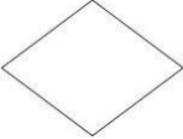
Tabel 2. 3 Simbol-Simbol Sequence Diagram

No.	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna,
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan

2.2.6 Entity Relationship Diagram (ERD)

Dalam rekayasa perangkat lunak diperlukan adanya ERD (*Entity Relationship Diagram*). ERD (*Entity Relationship Diagram*) merupakan model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Di dalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjuk objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya. Simbol-simbol ERD dapat di lihat pada Tabel 2.4 [9]

Tabel 2. 4 Simbol-simbol ERD

No	Simbol	Fungsi
1.	Entitas 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer.
2.	Relasi 	Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja.
3.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
4.	Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan.

5.	Atribut Multinilai 	<i>Field</i> yang butuh disimpan dalam entitas yang dapat memiliki nilai lebih dari satu.
4.	Alur 	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.

2.2.7 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Pemrograman berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek[13]. Berikut ini ada beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek yaitu :

1. Kelas (*Class*)

Kelas adalah dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/method), hubungan (relationship), dan arti. Suatu kelas dapat diwariskan ke kelas yang baru.

2. Operasi (*Method*)

Operasi atau metode atau method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi

yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

3. Atribut (*Property*)

Property atau disebut juga dengan atribut adalah variabel global yang dimiliki sebuah kelas. Atribut (*property*) dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privat untuk menjaga konsep enkapsulasi.

4. Objek

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

5. Abstraksi

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*Encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*Interface*)

Antarmuka atau interface sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka di mana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu.

9. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesifikasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

11. Komunikasi Antar Obyek

Komunikasi antar-objek dilakukan lewat *pesan (message)* yang dikirimkan dari satu objek ke objek lainnya.

12. Polimorfisme (*Polymorphism*)

Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. *Package*

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas – kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.8 Basis Data

Menurut Winarno pengertian basis data adalah : “Tempat untuk menyimpan berbagai data yang diperlukan oleh organisasi, perusahaan, instansi, pemerintahan, atau bahkan perorangan”. Basis data merupakan kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah maupun dimanipulasi untuk menghasilkan informasi. Basis data merupakan aspek yang sangat penting dalam sebuah sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. [10]

Dengan basis data seseorang dapat menyimpan sebuah informasi, seperti data perseorangan, mahasiswa, kepegawaian atau produk ke dalam media penyimpanan seperti *hardisk* melalui perangkat komputer, untuk kemudian data tersebut dapat digunakan sesuai keperluan.

2.2.9 MYSQL

MYSQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau yang biasa dikenal dengan DBMS (Database Management System), basis data ini *multi-thread* atau *multi-user*. SQL pertama kali didefinisikan oleh American National Standard Institute (ANSI) pada tahun 1986. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai Bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil. MySQL juga bersifat *open source* GPL (*General Public License*) mulai versi 3.23, pada bulan Juni 2000. [10]

MySQL merupakan sistem manajemen basis data yang bersifat *relational*, artinya data-data yang dikelola dalam basis data akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. MySQL dapat digunakan untuk mengelola basis data mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah Structured Query Language (SQL) untuk mengelola basis data-basis data yang ada di dalamnya.

2.2.10 C Sharp (C#)

C Sharp atau C# adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman *server-side* pada *website*, membangun aplikasi *desktop* ataupun *mobile*, pemrograman *game* dan sebagainya. [11]

C# sering dianggap sebagai bahasa penerus C++ atau versi canggih dari C++, karena ada anggapan bahwa # adalah perpaduan dari 4 buah tanda tambah yang disusun sedemikian rupa sehingga membentuk tanda pagar. Terlepas dari benar atau tidaknya anggapan tersebut, C# adalah sebuah bahasa pemrograman yang menjanjikan. C# merupakan bahasa pemrograman berbasis objek yang dikembangkan oleh *Microsoft* dan menjadi salah satu bahasa pemrograman yang mendukung *.NET programming* melalui Visual Studio.

2.2.11 Black Box Testing

Terfokus pada apakah unit program memenuhi kebutuhan (requirement) yang disebutkan dalam spesifikasi. Pada black box testing, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan.

Pengujian black box dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian black box harus dibuat dengan kasus benar dan kasus salah. Misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

- 1) Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang benar.
- 2) Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.2.12 Short Message Service (SMS)

Short Message Service SMS adalah Layanan yang banyak diaplikasikan pada jaringan komunikasi tanpa kabel yang memungkinkan dilakukannya pengiriman pesan dalam bentuk alphanumeric antar terminal pelanggan (ponsel) atau antara terminal pelanggan dengan sistem eksternal seperti e-mail, paging, voice mail, dan sebagainya. [12]

2.2.13 SMS Gateway

SMS *Gateway* merupakan gerbang bagi penyebaran informasi dengan menggunakan SMS. Anda dapat menyebarkan pesan ke banyak nomor secara otomatis dan cepat yang langsung terhubung dengan database nomor-nomor ponsel aja tanpa harus mengetik ratusan nomor dan pesan di ponsel anda, karena semua nomor akan diambil secara otomatis dari database tersebut. [12]

~Halaman Ini Sengaja Dikosongkan~

