BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

3.1.1 Sejarah Perusahaan

SMP AL-Hadi didirikan pada tahun 1984 dibawah Yayasan Pendidikan Islam H.Dull. Pada awal didirikannya sekolah ini hanya terdiri dari beberapa kelas dan murid saja. Jumlah gurunya pun belum terlalu banyak yang ditugaskan oleh yayasan. Seiring dengan berjalannya waktu, sekolah tersebut telah mengalami perkembangan yang begitu pesat yang dapat dilihat dari berkembangnya jumlah murid yang setiap tahunnya mengalami pertambahan yang pesat, begitu juga dengan jumlah gurunya yang mengalami pertambahan untuk mengimbangi jumlah siswa yang terdapat d SMP AL-Hadi. Dari awal didirikan SMP AL-Hadi telah menerapkan bimbingan baca tulis Al-Quran dan praktek ibadah. Sekolah tersebut didirikan berdasarkan atas rasa sosialisasi sebuah pendiri yayasan yang ingin mendirikan suatu sekolah islami yang dapat menghadapi tantangan global di masa depan.

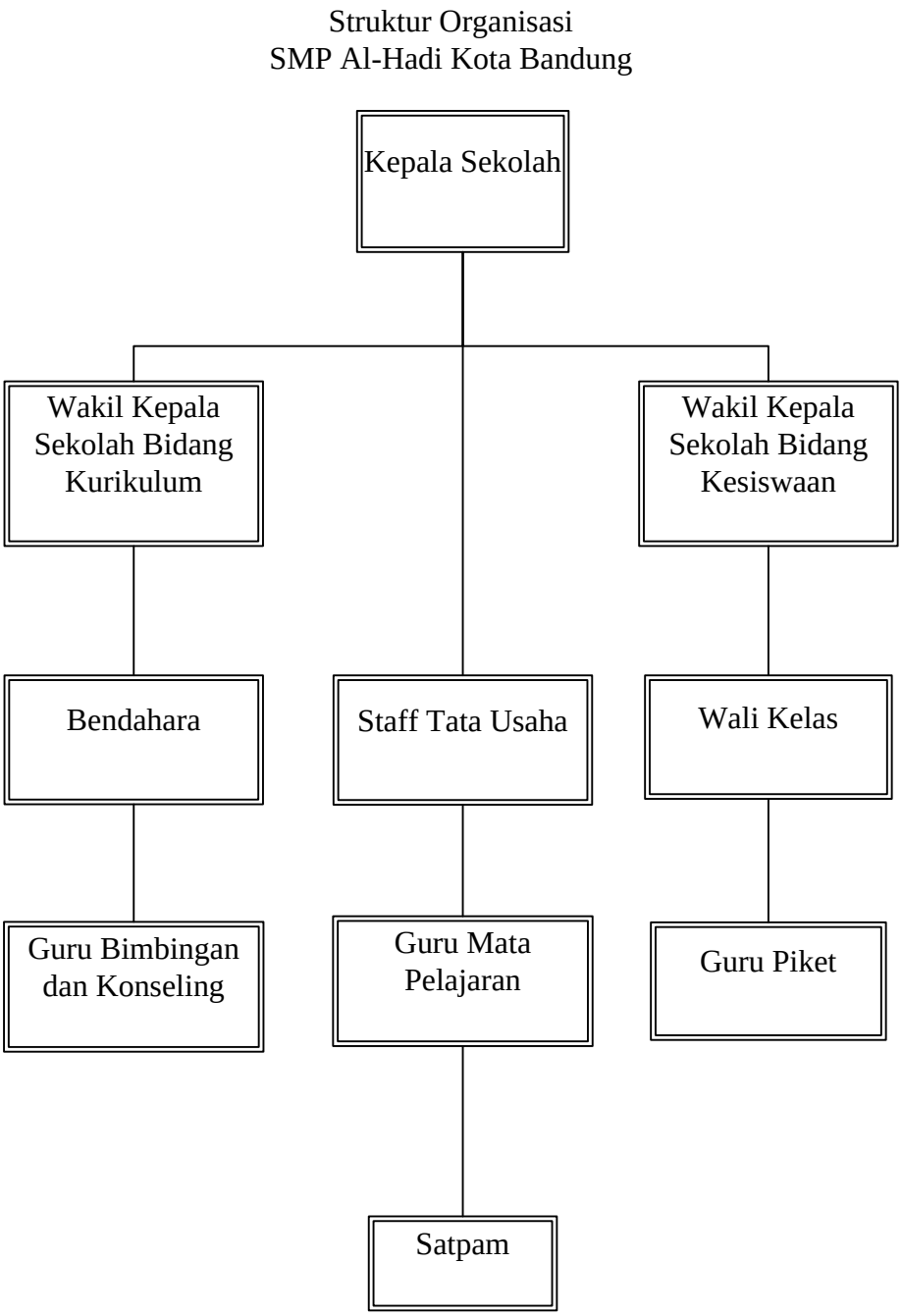
3.1.2 Visi dan Misi Perusahaan

Visi SMP AL-Hadi yaitu menjadi institusi pendidikan yang PASTI (Pelopor, Akhlakul Karimah, Sukses, Tanggap, Islami), unggul

dan siap menghadapi tantangan global. Adapun Misi SMP AL-Hadi adalah sebagai berikut :

1. Meningkatkan disiplin siswa, guru/tata usaha sehingga menjadi pelopor dalam kebaikan.
2. Menumbuh kembangkan sikap profesionalisme sehingga tercipta sumber daya manusia yang berakhlakul karimah, handal dan berkualitas.
3. Membekali siswa dengan ilmu pengetahuan dan keterempalian supaya menjadi orang yang sukses dimasa depan.
4. Menumbuh kembangkan sikap kritis dan tanggap terhadap permasalahan yang timbul.
5. Menciptakan lingkungan sekolah yang islami dan kondusif demi terciptanya suasana belajar yang menyenangkan.

3.1.3 Struktur Organisasi Perusahaan



Sumber: SMP Al-Hadi Kota Bandung

Gambar 3.1
Struktur Organisasi SMP AL-Hadi Kota Bandung

3.1.4 *Job Description*

Rincian tugas dari setiap bagian di SMP Al-Hadi Kota Bandung adalah sebagai berikut:

1. Kepala Sekolah

Kepala sekolah berfungsi dan bertugas sebagai Edukator, Manajer, Administrator dan Supervisor.

1. Sebagai Edukator bertugas melaksanakan proses belajar mengajar secara efektif dan efisien.
2. Sebagai Manajer mempunyai tugas menyusun perencanaan, mengorganisasikan, mengarahkan, mengkoordinasikan setiap kegiatan, melakukan evaluasi terhadap kegiatan, melaksanakan pengawasan, menentukan kebijakan, mengadakan rapat, mengambil keputusan, mengatur proses belajar mengajar, mengatur administrasi.
3. Sebagai Administrator mempunyai tugas membuat perencanaan, menyusun organisasi sekolah, sebagai coordinator dan pengarah, melaksanakan pengelolaan kepegawaian.
4. Sebagai Supervisor bertugas dan bertanggung jawab apabila hendak meneliti dan menentukan syarat-syarat apa saja yang diperlukan bagi kemajuan sekolahnya sehingga tujuan sekolah dapat tercapai.

2. Wakil Kepala Sekolah Bidang Kurikulum.

Wakil kepala sekolah bidang kurikulum memiliki tugas sebagai berikut:

1. Menyusun dan menjabarkan kalender pendidikan.
2. Menyusun pembagian tugas guru dan jadwal pelajaran.
3. Mengatur penyusunan program pengajaran.
4. Mengatur pelaksanaan program penilaian criteria kenaikan kelas, kelulusan dan laporan kemajuan belajar siswa serta pembagian raport dan ijazah.
5. Mengatur pelaksanaan program perbaikan dan pengajaran.
6. Memantau dan menerima laporan kinerja guru dan absensi kehadiran guru.
7. Mengatur mutasi siswa.
8. Melakukan supervise administrasi dan akademis.
9. Menyusun laporan.

3. Wakil Kepala sekolah Bidang Kesiswaan.

   Wakil kepala sekolah bidang kesiswaan memiliki tugas sebagai berikut:

   1. Mengatur program dan pelaksanaan bimbingan dan konseling.
   2. Mengatur dan membina program kerja OSIS.
   3. Mengatur program pesantren kilat.
   4. Menyusun dan mengatur pelaksanaan pemilihan siswa teladan sekolah.
   5. Menyelenggarakan cerdas cermat dan olahraga prestasi.
   6. Menyeleksi calon untuk diusulkan mendapat beasiswa.
   7. Menyusun laporan.

4. Bagian Bendahara

   Bendahara sekolah memiliki tugas sebagai berikut:

   1. Mengatur biaya untuk kebutuhan sekolah.
   2. Mengatur biaya untuk penggajian guru dan pegawai sekolah.
   3. Pengelola keuangan sekolah.

5. Guru Mata Pelajaran.

   Guru memiliki tugas sebagai berikut:

   1. Membuat perangkat proses pengajaran.
   2. Melaksanakan kegiatan pembelajaran.
   3. Melaksanakan kegiatan penilaian proses belajar.
   4. Melaksanakan analisis hasil ulangan.
   5. Mengisi daftar nilai siswa.
   6. Mengadakan pengembangan program yang menjadi tanggung jawabnya.
   7. Membuat catatan tentang hasil kemampuan belajar siswa.

6. Wali Kelas.

   Wali Kelas memiliki tugas sebagai berikut:

   1. Pengelolaan kelas.
   2. Penyelenggaraan administrasi kelas.
   3. Pengisian daftar keseluruhan nilai siswa.
   4. Pembuatan catatan khusus tentang siswa.
   5. Pencatatan mutasi siswa.
   6. Pengisian Buku Laporan Penilaian Hasil Belajar.

7. Pembagian Buku Laporan Penilaian Hasil Belajar.

7. Guru Bimbingan dan Konseling.

   Guru Bimbingan dan Konseling memiliki tugas sebagai berikut:

   1. Penyusunan program dan pelaksanaan bimbingan dan konseling.
   2. Koordinasi dengan wali kelas dalam rangka mengatasi masalah-masalah yang dihadapi oleh siswa tentang kesulitan belajar.
   3. Mengadakan penilaian pelaksanaan bimbingan dan konseling.

8. Tata Usaha.

   Tata Usaha memiliki tugas sebagai berikut:

   1. Penyusunan program kerja tata usaha sekolah.
   2. Penyusunan administrasi perlengkapan sekolah.
   3. Penyusunan dan penyajian data statistic sekolah.
   4. Penerima laporan akhir atau arsip semua bagian.
   5. Penyusunan laporan kegiatan ketata usahaan secara berkala.

9. Guru Piket.

   Guru Piket memiliki tugas sebagai berikut:

   1. Menjaga memantau pelaksananan kegiatan belajar mengajar.
   2. Memantau kegiatan dan disiplin guru dalam hal kehadiran setiap harinya.

10. Satpam

    Satpam memiliki tugas untuk menjaga, mengamankan dan mengawasi sekolah.

3.2 Metode Penelitian

Metode penelitian merupakan suatu mekanisme, teknik atau cara untuk mencari, memperoleh, mengumpulkan, atau mencatat data yang dapat digunakan untuk menyusun karya ilmiah atau penelitian dengan prosedur yang didasarkan pada suatu struktur logis yang terdiri dari beberapa tahapan kerja dan kemudian menganalisa faktor-faktor yang berhubungan dengan pokok-pokok permasalahan sehingga akan didapat suatu kebenaran atas data yang diperoleh.

3.2.1 Desain Penelitian

Dalam membangun Sistem Informasi yang kompleks membutuhkan metode-metode atau paradigma pengembangan yang mampu membantu menganalisis dan mendesain secara lebih detail sehingga informasi yang di hasilkan lebih akurat.

Jenis metode penelitian yang digunakan penulis dalam penelitian ini adalah metode deskriptif dan *action research*. Metode deskriptif yaitu metode dalam penelitian suatu kasus dengan cara mengumpulkan data sebagai gambaran keaadaan objek yang diteliti berdasarkan fakta-fakta yang ada. Teknik Analisis Data Metode Deskriptif menurut Sugiyono (2005:21) adalah :

> "Metode deskriptif adalah suatu metode yang digunakan untuk menggambarkan atau menganalisis suatu hasil penelitian tetapi tidak digunakan untuk membuat kesimpulan yang lebih luas".

Metode deskriptif ini digunakan untuk menguji dan mengetahui sifat serta hubungan yang lebih mendalam antara dua variabel dengan cara mengamati aspek-aspek tertentu secara lebih spesifik untuk memperoleh data yang sesuai dengan masalah yang ada dengan tujuan penelitian, dimana data tersebut diolah, dianalisis, dan diproses lebih lanjut dengan dasar teori-teori yang telah dipelajari sehingga data tersebut dapat ditarik sebuah kesimpulan.

Sedangkan *action research* (tindakan) yaitu mengembangkan keterampilan-keterampilan baru, cara pendekatan baru, atau produk pengetahuan baru dan memecahkan masalah dengan penerapan langsung di dunia aktual / lapangan.

3.2.2 Jenis dan Metode Pengumpulan Data

Dalam pengembangan sistem informasi tentunya diperlukan jenis data yang akurat sesuai dengan sistem yang akan dikembangkan, oleh karena itu dibutuhkan suatu teknik pengumpulan data yang tepat. Jenis data yang digunakan penulis dalam tahap analisa dan tahap perancangan adalah sebagai berikut

3.2.2.1 Sumber Data Primer

Data primer adalah data yang dikumpulkan oleh penulis secara langsung melalui objek penelitian, yaitu dengan cara observasi, wawancara, mengamati dan mencatat.

a. Wawancara (Interview)

Penulis melakukan wawancara pada staf tata usaha (TU) dan mewawancarai mengenai system absensi dan penggajian di SMP AL-Hadi Kota Bandung.

Wawancara merupakan salah satu usaha atau percobaan secara sistematis, untuk mengumpulkan informasi yang kita butuhkan dari seseorang. Tujuan dari wawancara adalah untuk memperoleh informasi yang lebih akurat dan lengkap, untuk menyusun sistem yang baru agar sesuai dengan kebutuhan sistem organisasi.

b. Pengamatan (Observasi)

Observasi yaitu dengan meninjau dan mengamati secara langsung sistem yang sedang berjalan di perusahaan tersebut serta mengumpulkan data atau informasi yang terkait dengan sistem-sistem informasi penggajian yang selanjutnya akan dianalisis dalam analisa sistem berjalan.

Disini penulis mengumpulkan data-data melalui staf tata usaha dan penulis juga mengamati sistem di sekolah tersebut.

3.2.2.2 Sumber Data Sekunder

Data sekunder merupakan cara pengumpulan data dengan cara mempelajari data yang telah tersedia atau dikumpulkan. Cara yang digunakan untuk mengumpulkan data sekunder adalah metode Dokumentasi, yaitu dengan mengumpulkan data dan informasi yang diperlukan dari sumber-sumber kebanyakan dari materi sejenis dokumen

yang berkenaan dengan masalah yang diteliti. Metode ini digunakan untuk pengumpulan data yang berhubungan dengan sejarah, tujuan, kegiatan dan struktur organisasi.

3.2.3 Metode Pendekatan dan Pengembangan Sistem

Adapun metode pendekatan dan pengembangan sistem akan dijelaskan seperti dibawah ini antara lain yaitu:

3.2.3.1 Metode Pendekatan Sistem

Pendekatan Sistem adalah upaya untuk melakukan pemecahan masalah yang dilakukan dengan melihat masalah yang ada secara menyeluruh dan melakukan analisis secara sistem.

Langkah-langkah yang harus dilakukan dalam suatu analisa sistem yang baik adalah :

1. Tentukan *input* dan *output* dasar dari sistem.
2. Tentukan proses yang dilakukan di tiap-tiap tahap.
3. Rancang perbaikan sistem dan lakukan pengujian dengan :
    a. *Fersibility* : cari yang memungkinkan.
    b. *Viability* : kelangsungan.
    c. *Cost* : cari yang harganya murah/terjangkau.
    d. *Effectiveness* : dengan *input* yang sedikit, *output* besar.
4. Buat rencana kerja dan penunjukkan tenaga.
5. Implementasikan dan penilaian terhadap sistem yang baru.

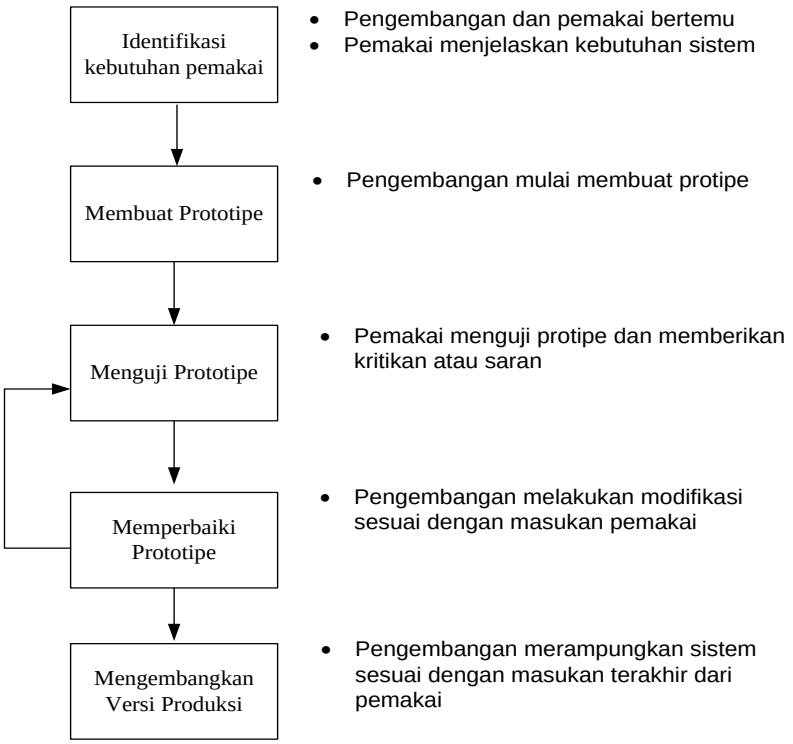
3.2.3.2 Metode Pengembangan Sistem

Pengembangan Sistem yang penulis gunakan dalam Skripsi ini adalah dengan menggunakan metode pengembangan prototipe. Menurut Abdul Kadir prototipe adalah suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat di evaluasi oleh pemakai.(Kadir, 2003:416).

Secara garis besar, menurut Lucas dalam Abdul Kadir sasaran prototipe adalah sebagai berikut:

1. Mengurangi waktu sebelum pemakai melihat sesuatu yang konkret dari usaha pengembangan sistem.
2. Menyediakan umpan balik yang cepat dari pemakai kepada pengembang.
3. Membantu menggambarkan kebutuhan pemakai dengan kesalahan yang lebih sedikit.
4. Meningkatkan pemahaman pengembang dan pemakai terhadap sasaran yang harus dicapai oleh sistem.
5. Menjadikan keterlibatan pemakai sangat berarti dalam analisis dan desain sistem.(Lucas dalam Kadir, 2003:416).

Pendekatan prototipe menurut Abdul Kadir dapat digambarkan sebagai berikut:



Sumber: (Kadir, 2003:417).

Gambar 3.2
Mekanisme Pengembangan Sistem Dengan Prototipe

3.2.3.3 Alat Bantu Analisis dan Perancangan

Alat bantu analisis yang digunakan dalam perancangan sistem ini adalah dengan menggunakan metode terstruktur. Alat bantu perancangan sistemnya dengan menggunakan *Flow Map*, Diagram Konteks, *Data flow Diagram* (*DFD*), Kamus Data, *Entity Relation Diagram (ERD)*, dan tabel relasi.

1. Flowmap

Digunakan untuk mendefinisikan hubungan antara bagian (*Pelaku Proses*), proses (*manual atau berbasis komputer*) dan aliran data dalam

bentuk dokumen keluaran dan masukan. Sedangkan definisi dari simbol flow map diantaranya :

a. Simbol Dokumen : menunjukan dokumen input dan output baik untuk proses manual maupun computer.
b. Simbol kegiatan manual : menunjukan kegiatan manual.
c. Simbol simpanan offline (bukan computer).
d. Simbol proses : menunjukan kegiatan proses operasi komputer.
e. Simbol keputusan : digunakan untuk menyeleksi proses.
f. Simbol penghubung : menunjukan penghubung ke halaman yang sama atau lainnya.
g. Simbol terminator : untuk menyatakan awal dan akhir proses.
h. Simbol garis : menunjukan arus dari proses.

2. Diagram Konteks

Pengertian diagram kontek adalah suatu diagram alir yang tingkat tinggi yang menggambarkan seluruh jaringan, masukan dan keluaran. sistem yang dimaksud adalah untuk menggambarkan sistem yang sedang berjalan. mengidentifikasikan awal dan akhir data awal dan akhir yang masuk dan keluaran sistem.

Diagram ini merupakan gambaran umum sistem yang nantinya akan kita buat. secara uraian mah dapat dikatakan bahwa diagram kontek itu berisi siapa saja yang memberikan data (inputan) kesistem serta kepada siapa data informasi yang harus dihasilkan sistem.

3. *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan sistem yang sedang berjalan logis. Dalam sumber lain dikatakan bahwa DFD ini merupakan salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Suatu yang lazim bahwa ketika menggambarkan sebuah sistem kontekstual *data flow diagram* yang akan pertama kali muncul adalah interaksi antara sistem dan entitasluar. DFD didesain untuk menunjukkan sebuah sistem yang terbagi-bagi menjadi suatu bagian sub-sistem yang lebih kecil adan untuk menggaris bawahi arus data antara kedua hal yang tersebut diatas. Diagram ini lalu "dikembangkan" untuk melihat lebih rinci sehingga dapat terlihat model-model yang terdapat di dalamnya.

DFD merupakan alat yang cukup populer, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. Berikut simbol-simbol yang digunakan untuk menggambarkan diagram arus data menurut Jogiyanto HM (2005 : 700).

1. *External Entity* (Kesatuan Luar)

   Setiap sistem pasti memiliki suatu batasan sistem yang memisahkan sistem dengan lingkungan luarnya dimana akan menerima input dan menghasilkan output. Kesatuan luar dapat berupa orang, organisasi atau sistem lainnya.

2. *Data Flow* (Arus Data)

   Arus data diberi simbol panah yang menunjukkan arus dari suatu data yang dapat berupa masukkan atau hasil dari suatu proses, arus data mengalir di antara proses, simpanan data dan kesatuan luar.

3. *Proccess* (Proses)

   Proses merupakan kegiatan yang dilakukan oleh orang atau komputer dari hasil suatu arus data yang masuk ke dalam proses. Untuk dihasilkan arus data yang keluar dari proses.

4. *Data Store* (Simpanan Data)

   Simpanan data adalah simpanan dari data yang dapat berupa file atau database sistem komputer atau yang bersifat manual seperti buku, alamat, atau folder.

4. Kamus Data

Kamus data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir berisi tentang struktur database. Kamus data biasanya dipelihara secara otomatis oleh sistem manajemen database. Berikut ini notasi yang akan dipakai dalam kamus data:

Tabel 3.1 Notasi Kamus Data

Notasi	Arti
=	Terdiri dari atau sama dengan
+	Dan
( )	Optional (Elemen data dalam tanda kurung biasa ada bisa tidak)
{ }	Iteras
[ ]	Pilih salah satu dari beberapa pilihan
*	Komentar atau keterangan
@	Identitas atau field kunci
|	Sama dengan tanda [ ]

5. Perancangan Basis Data

Perancangan basis data adalah proses pembuatan (develop) stuktur database sesuai dengan data yang dibutuhkan oleh user.

a. Normalisasi.

Normalisasi merupakan suatu proses pengelompokan data elemen menjadi tabel-tabel yang menunjukan entity dan relasi yang berfungsi untuk menghilangkan adanya redudansi data dan menentukan key unik untuk mengakses data atau merupakan pembentukan relasi sedemikian rupa sehingga database tersebut mudah dimodifikasi.

b. Tabel Relasi

Tabel relasi berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari "dunia nyata" yang dapat digambarkan dengan lebih sistematis dengan menggunakan ERD.

3.2.4 Pengujian Software

Pengujian *Software* adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Dalam pengujian perangkat lunak dilakukan pengujian melalui beberapa tingkatan, dikenal ada empat (4) tingkatan utama dalam melakukan pengujian terhadap perangkat lunak yang kita bangun. Tingkatan tersebut adalah sebagai berikut:

1. Pengujian Unit, Form atau Modul
2. Pengujian Integrasi
3. Pengujian Validasi
4. Pengujian Sistem

Sementara itu Perangkat lunak yang kita bangun dapat diuji dengan dua cara, yaitu:

1. Pengujian dengan menggunakan data uji untuk menguji semua elemen program (data internal, *loop*, logika, keputusan dan jalur). Data uji

dibangkitkan dengan mengetahui struktur internal (kode sumber) dari perangkat lunak.

2. Pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional perangkat lunak bekerja dengan baik. Data uji dibangkitkan dari spesifikasi perangkat lunak.

3.2.4.1 *Black Box Testing*

Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah sesuai dengan yang diharapkan. *Black box* berusaha menemukan kesalahan dalam kategori:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Black-Box Testing terfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Black Box

Testing bukanlah solusi alternatif dari White-Box Testing tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh White-Box Testing.

3.2.4.2 Jenis-Jenis Black Box Testing

Dalam metode *black-box testing* terdapat jenis-jenis metode black-box testing yang dapat membantu melakukan pengujian sistem. Berikut ini adalah jenis-jenis metode black-box testing:

1. Equivalence Partitioning

Equivalence Partitioning membagi input menjadi kelas-kelas data yang dapat digunakan untuk menggenerasi kasus uji, bertujuan untuk menemukan kelas-kelas kesalahan dan berdasarkan pada kesamaan kelas-kelas kondisi input. Sebuah kelas yang ekivalen merepresentasikan kumpulan status/kondisi yang valid atau tidak valid.

2. Boundary Value Analysis / Limit Testing

Banyak kesalahan terjadi pada kesalahan masukan. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input. BVA merupakan komplemen dari *equivalence partitioning*. Lebih pada memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas.

3. Comparison Testing

Pada beberapa aplikasi *reliability* dari sebuah perangkat lunak sangat penting. Redundansi perangkat keras dan perangkat lunak mungkin

digunakan untuk meminimalisir kesalahan *(error)*. Untuk redundansi perangkat lunak, gunakan tim yang terpisah untuk mengembangkan setiap versi perangkat lunak yang independen. Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama. Jalankan semua versi dengan paralel dan perbandingan keluaran secara *real-time*. Walau hanya dijalankan sebuah versi pada akhirnya, untuk beberapa aplikasi yang penting dapat mengembangkan versi independen dan menggunakan *comparison testing* atau *back-to-back testing*. Ketika output dari versi berbeda, maka setiap versi diinvestigasi jika ada kemungkinan *defect.* Metode ini tidak untuk menemukan kesalahan dari spesifikasi.

4. Sample Testing

Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen. Mengintegrasikan nilai pada kasus uji. Nilai-nilai yang terpilih mungkin dipilih dengan urutan tertentu atau interval tertentu.

5. Robustness Testing

Data input dipilih diluar spesifikasi yang telah didefinisikan. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.

6. Behavior Testing

Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack.

7. Performance Testing

Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi, dll. Untuk mencari tahu beban kerja atau kondisi konfigurasi program. Spesifikasi mengenai performansi didefinisikan pada saat tahap spesifikasi atau desain. Dapat digunakan untuk menguji batasan lingkungan program.

8. Requirement Testing

Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (input/output/fungsi/performansi) diidentifikasi pada tahap spesifikasi kebutuhan dan desain. Requirement testing melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program. Untuk memfasilitasinya, setiap spesifikasi kebutuhan bisa ditelusuri dengan kasus uji dengan menggunakan *traceability matrix*.

9. Endurance Testing

Endurance Testing melibatkan kasus uji yang diulangulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai dengan spesifikasi kebutuhan.

10. Cause-effect Relationship Testing

Teknik ini merupakan suplemen dari *equivalence testing* dengan menyediakan cara untuk memilih kombinasi data input. Melibatkan kondisi input *(cause)* dan kondisi output *(effect)* untuk mencegah pendefinisian kasus uji yang terlalu banyak langkah. Bagi-bagi spesifikasi kebutuhan

menjadi bagian yang memiliki kemungkinan kerja. Definisikan *cause* dan *effect* berdasarkan spesifikasi kebutuhan.

3.3 Pengenalan PHP

3.3.1 Pengertian PHP

PHP adalah bahasa pemrograman script yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Aplikasi bahasa PHP dapat digunakan sebagai landasan operasi pada pemrogaman jaringan berbasis web. Selain itu PHP juga bisa digunakan untuk membuat program database dan utnuk membuat aplikasi web3.

3.3.2 Sejarah PHP

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang PHP: Hypertext Preprocessing.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

3.3.3 Kelebihan PHP

PHP memiliki beberapa kelebihan yang membuat para pengguna tertarik untuk memakai bahasa pemograman tersebut. Kelebihan PHP adalah sebagai berikut:

1. PHP menjadi popular karena kesederhanaannya dengan kemampuannya dakam menghasilkan berbagai aplikasi web seperti counter, system artikel atau biasa disebut dengan CMS, *e-commerce*, bulletin board, dsb.
2. PHP adalah salah satu bahasa server-side yang didesain khusus untuk aplikasi web.
3. PHP termasuk dalam Open Source Product dan telah mencapai versi 4.
4. Aplikasi PHP cukup cepat dibandingkan dengan aplikasi CGI dengan Perl atau Phyton bahkan lebih cepat dibandingkan dengan ASP maupun Java dalam berbagai aplikasi web.
5. Tersedia baik di Windows maupun Linux, walau saat ini paling efektif di web server Apache dan OS Linux.
6. Sintaks mirip C dan mudah dipelajari.

3.3.4 Kekurangan PHP

Selain memiliki kelebihan, PHP juga memiliki kekurangan yaitu sebagai berikut:

1. Tidak detail untuk pengembangan skala besar.
2. Tidak memiliki system pemrogaman berorientasi objek yang sesungguhnya.
3. Tidak bisa memisahkan antara tampilan dengan logic dengan baik.

4. PHP memiliki kelemahan security tertentu apabila programmer tidak jeli dalam melakukan pemograman dan kurang memperhatikan isu konfogurasi PHP.
5. Kode PHP dapat dibaca semua orang, dan kompilasi hanya dapat dilakukan dengan tool yang mahal dari Zend.

3.4 Pengenalan Gammu

Gammu adalah sebuah aplikasi yang dapat digunakan untuk mengelola berbagai fungsi pada handphone, modem dan perangkat sejenis lainnya. Fungsi-fungsi yang dapat dikelola Gammu antara lain adalah fungsi nomor kontak (phonebook) dan fungsi SMS. Namun, untuk aplikasi yang akan saya coba ulas di artikel ini akan lebih banyak menggunakan fungsi SMS dari Gammu. Kelebihan Gammu dibandingkan tool SMS Gateway lainnya adalah sebagai berikut:

a. Gammu bisa dijalankan Windows ataupun Linux.
b. Banyak device yang compatible dengan Gammu.
c. Gammu menggunakan database MySQL.
d. Baik kabel data USB maupun serial, semuanya kompatibel dengan Gammu.
e. Gammu adalah aplikasi open source yang dapat dipakai secara gratis.
f. Gammu tidak memerlukan banyak hardware (hanya memerlukan PC dan Modem), sehingga memudahkan dalam mengembangkan aplikasi dengan terjangkau.