BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Pada penulisan skripsi ini penulis melakukan penelitian di SMP Al-Hadi Kota Bandung. Adapun yang akan dibahas pada objek penelitian ini yaitu mengenai sejarah singkat SMP Al-Hadi Kota Bandung, visi dan misi SMP Al-Hadi Kota Bandung, struktur organisasi dan deskripsi tugas yang terdapat pada SMP Al-Hadi Kota Bandung.

3.1.1 Sejarah Singkat Perusahaan

SMP Al-Hadi Kota Bandung 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 Kota Bandung. Dari awal didirikan SMP Al-Hadi Kota Bandung 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.

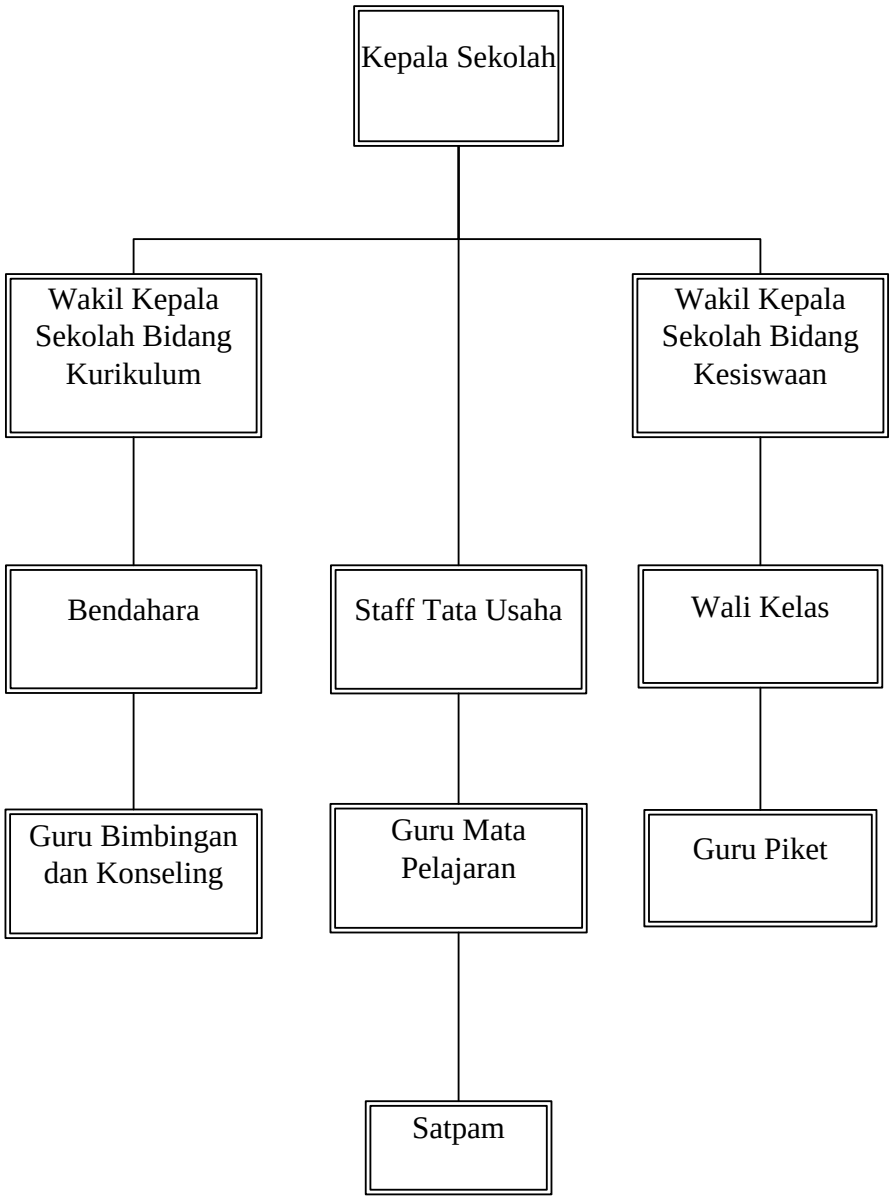
3.1.2 Visi dan Misi Perusahaan

Visi SMP Al-Hadi Kota Bandung yaitu menjadi institusi pendidikan yang PASTI (Pelopor, Akhlakul Karimah, Sukses, Tanggap, Islami). Adapun Misi SMP Al-Hadi Kota Bandung adalah:

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

Struktur Organisasi
SMP Al-Hadi Kota Bandung



Sumber : SMP Al-Hadi Kota Bandung, 2012

Gambar 3.1
Struktur Organisasi SMP AL-Hadi Kota Bandung

3.1.4 Deskripsi Tugas

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 adalah salah suatu teknis dan cara mencari, memperoleh, mengumpulkan dan mencatat data, baik berupa primer maupun data sekunder yang di gunakan untuk keperluan menyusun suatu karya ilmiah. Metode

penelitian yang dilakukan oleh penulis adalah metode deskriptif dan metode action.

Pengertian metode deskriptif menurut Sugiyono adalah sebagai berikut:

> "Metode deskriptif adalah metode yang digunakan untuk menggambarkan atau menganalisis suatu penelitian tetapi tidak digunakan untuk membuat kesimpulan yang lebih luas". (Sugiyono, 2005: 21).

Metode penelitian yang digunakan adalah pendekatan deskriptif analis dengan pendekatan kualitatif, yaitu suatu metode yang berusaha memberikan gambaran mengenai data atau kejadian berdasarkan fakta-fakta yang tampak pada situasi yang diselidiki peneliti dan objek yang diteliti terpisah, proses penelitian yang dilakukan melalui pengukuran dengan alat yang baku dan objektif.

Metode yang dilakukan oleh penulis bukan hanya metode deskriptif namun digunakan juga metode action. Metode action atau tindakan merupakan penelitian dimana penulis berupaya untuk memecahkan masalah dalam dunia nyata dengan mengkaji pengalaman-pengalaman dalam memecahkan masalah tersebut.

3.2.1 Desain Penelitian

Dalam melakukan suatu penelitian sangat perlu dilakukan perencanaan dan perancangan penelitian, agar penelitian yang dilakukan dapat berjalan dengan baik dan sistematis.

Desain penelitian merupakan rancangan penelitian yang digunakan sebagai pedoman dalam melakukan proses penelitian. Desain penelitian akan

berguna bagi semua pihak yang terlibat dalam proses penelitian, karena langkah, dalam melakukan penelitian mengaju kepada desain penelitian yang telah dibuat.

Menurut Moh. Nazir dalam bukunya *Metodologi Penelitian* mengemukakan bahwa :

> "Desain penelitian adalah Desain penelitian merupakan rancangan utama penelitian yang menyatakan metode dan prosedur-prosedur yang digunakan oleh peneliti dalam pemilihan, pengumpulan, dan analisis data.*"*(Nazir, 2005:84).

Dari pemaparan diatas, maka dapat dikatakan bahwa desain penelitian merupakan semua proses penelitian yang dilakukan oleh penulis dalam melaksanakan penelitian mulai dari perencanaan sampai dengan pelaksanaan penelitian yang dilakukan pada waktu tertentu.

Dalam penelitian ini, penulis menggunakan desain penelitian dengan langkah- langkah adalah sebagai berikut:

1. Penelitian ini dilakukan karena adanya masalah yang terdapat pada sekolah SMP Al-Hadi mengenai pengolahan data akademik. Dengan adanya masalah tersebut penulis ingin memecahkan masalah tersebut melalui sebuah penelitian. Berdasarkan masalah penulis menetapkan judul yang diteliti, sehingga dapat diketahui apa yang akan diteliti. Dalam penelitian ini penulis mengambil judul "Sistem Informasi Akademik Berbasis SMS *gateway* SMP Al-Hadi Kota Bandung".

2. Merumuskan masalah penelitian termasuk membuat spesifikasi dari tujuan, dan batasan masalah. Masalah yang diteliti dalam penelitian ini adalah pengolahan data akademik.
3. Mengamati, mengumpulkan, melakukan wawancara serta menganalisis data-data yang diperlukan mengenai proses pengolahan data akademik yang terdapat pada SMP Al-Hadi Kota Bandung.
4. Melaporkan hasil dari penelitian dan mengajukan beberapa saran untuk kemajuan SMP Al-Hadi Kota Bandung.

3.2.2 Jenis dan Metode Pengumpulan Data

Didalam penelitian yang dilakukan oleh penulis dibutuhkan suatu metode yang digunakan sebagai alat atau sarana pengambilan data-data, metode yang dilakukan adalah sebagai berikut :

3.2.2.1 Sumber Data Primer

Sumber data primer adalah sumber data yang didapatkan langsung dari orang yang mengerti tentang permasalahan tersebut yaitu dengan dilakukannya wawancara, observasi dan kuisioner. Di bawah ini metode sumber data primer yang penulis lakukan:

1. **Observasi**

Observasi yaitu teknik pengumpulan data dengan cara penulis mengamati data dan informasi yang diharapkan sehingga penulis dapat mengetahui bagaimana proses pengolahan data nilai siswa yang dilakukan oleh SMP Al-Hadi Kota Bandung.

2. Wawancara

Wawancara yaitu cara memperoleh informan melalui komunikasi percakapan yang dilakukan saling berhadapan ataupun langsung maupun melalui telepon. Peneliti mewawancarai guru dan staff TU yang berada di SMP Al-Hadi Kota Bandung, yang mengetahui dan memahami lebih jauh mengenai pengolahan data akademik agar diperoleh data yang akurat.

3.2.2.2 Sumber Data Sekunder

Data sekunder yang didapat oleh penulis yaitu berupa dokumentasi. Dokumentasi yaitu data yang telah di olah oleh instansi tersebut dan telah dilakukan penelaahan melalui dokumen tertulis yang diperoleh dari instansi terkait berupa cuplikan, kutipan, atau penggalan dari catatan organisasi.

3.2.3 Metode Pendekatan dan Pengembangan Sistem

Dalam pemahaman masalah dilakukan pendekatan sistem yang merupakan serangkaian langkah-langkah pemecahan masalah yang memastikan bahwa masalah dipahami, solusi *alternative* dipertimbangkan dan solusi yang dipilih bekerja. Kemudian akan dilanjutkan dengan pengembangan sistem sebagai bentuk aktivitas untuk menghasilkan sistem informasi berbasis komputer untuk menyelesaikan persoalan organisasi atau memanfaatkan kesempatan yang timbul.

3.2.3.1 Metode Pendekatan Sistem

Metode pendekatan sistem yang digunakan oleh penulis adalah dengan menggunakan metode pendekatan terstruktur. Metode pendekatan terstruktur

adalah suatu proses untuk mengimplementasikan langkah dalam menyelesaikan suatu masalah dalam bentuk program. Pemograman terstruktur adalah suatu proses pengimplentasian urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program.

Pendekatan terstruktur membutuhkan suatu alat bantu atau *tools* dalam pengembangan sistem, di mana dalam metode terstruktur menggunakan alat bantu analisis seperti *Flow Map,* Diagram Konteks, *Data Flow Diagram* (DFD), Kamus Data, Normalisasi, Relasi tabel, dan E*ntity Relationship Diagram* (ERD) sehingga dapat menghasilkan suatu sistem yang dikembangkan menjadi sistem yang strukturnya dapat didefinisikan dengan baik, jelas dan dapat menghasilkan suatu tujuan yang hendak dicapai.

3.2.3.2 Metode Pengembangan Sistem

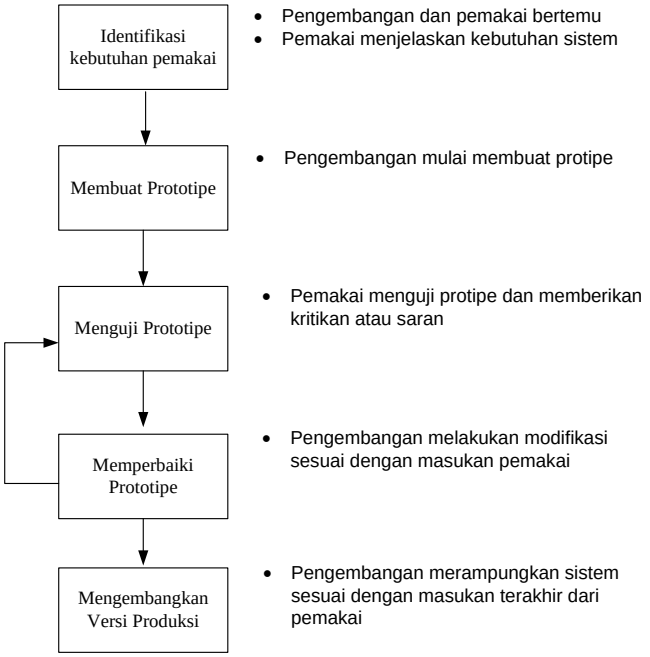
Pengembangan Sistem dalam proposal usulan penelitian ini menggunakan metode pengembangan sistem menggunakan metode prototipe. Menurut Abdul Kadir

> ”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:



Gambar 3.2
Mekanisme pengembangan sistem dengan prototype
Sumber: (Kadir, 2003:417).

Dimana tahapan-tahapan yang harus dilaksanakan dalam pengembangan prototype adalah sebagai berikut :

1. Identifikasi kebutuhan sistem

Pada tahap ini merupakan tahap awal dalam membangun sebuah sistem informasi, dimana antara pemakai sistem (user) dan pengembang sistem bertemu. user menjelaskan tentang kebutuhan sistem yang akan dibangun oleh pengembang sistem dan pengembang melihat beberapa kebutuhan yang dibutuhkan serta berusaha membuat sesuai dengan kebutuhan.

2. Membuat prototype

Setelah menganalisa sistem yang akan di kembangkan serta kebutuhan kebutuhan sistem sistem yang akan dibangun, pengembang sistem mulai membuat prototype.

3. Menguji prototype

User melakukan pengujian program agar program dapat digunakan sesuai dengan kebutuhan, dan *user* memberikan saran atau masukan bila terdapat kekurangan pada program.

4. Memperbaiki prototype

Di sini pembangun sistem melakukan perbaikan/penambahan pada program yang di bangun, jika terdapat kesalahan yang terjadi pada sistem. Kesalahan didapat sesuai dengan uji coba yang dilakukakan oleh user dan pembangun sistem. Setelah memahami kesalahan atau kekurangan maka dilakukan perbaikan prototype.

5. Mengembangankan versi produksi

Pada tahap ini pembangun sistem menyelesaikan sistem yang telah dibuatnya sesuai dengan masukan atau saran terakhir dari pemakai sistem dan melakukan perbaikan sistem yang mengalami kesalah dan dibuatkannya kembali sistem yang terbaru sesuai dengan kebutuhan user.

Model prototype memiliki beberapa keunggulan dalam pembangunan sebuah sistem, berikut ini adalah keunggulan dari penggunaan model prototype:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan.
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
3. Pelanggan berperan aktif dalam pengembangan sistem.
4. Lebih menghemat waktu dalam pengembangan sistem.
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Model prototype juga memiliki kelemahan dalam pengembangan sistem, berikut ini adalah beberapa kelemahan dari model prototype :

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk

membuat prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem .

3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik.

3.2.3.3 Alat Bantu Analisis dan Perancangan

3.2.3.3.1 Flowmap

Menurut Andri Kristato (2003:68) flowmap atau dapat disebut juga block chart atau flowchart berfungsi untuk memodelkan masukan atau keluaran proses maupun transaksi dengan simbol – simbol tertentu (Andri Kristanto, 2003:68).

Suatu flow map digambarkan sebagai pemetaan hubungan antara bagian-bagian kerja melalui dokumen, baik berupa laporan maupun formulir. Flow map digunakan untuk menganalisis bagaimana hubungan antara sub kerja yang akan menggerakkan sistem. Setelah diketahui bagian-bagian yang terlibat dalam sistem, maka akan diketahui berapa jumlah entitas yang terkait dengan sistem yang dianalisis dan dirancang.

Flowmap adalah suatu alat untuk membantu seorang analis dan programmer untuk memecahkan suatu masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis suatu pengoperasian. Ada beberapa petunjuk yang harus diperhatikan, seperti :

1. *Flow map* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.

3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
5. Lingkup dan range dari aktvitas yang sedangdigambarkan harus ditelusuri dengan hati-hati.
6. Gunakan simbol-simbol flowmap yang standar.

Dari pendapat diatas penulis dapat menarik kesimpulan bahwa definsi dari *flowmap* adalah suatu bagan aliran data untuk menunjukan suatu aliran data pada program atau prosedur secara terstruktur untuk menyelesaikan suatu masalah.

3.2.3.3.2 Diagram Kontek

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.

Pengertian lain dijelaskan oleh Sutabari dalam bukunya yang berjudul *Analisis Sistem Informasi,* definisi diagram konteks adalah sebagai berikut:

> "Diagram konteks dibuat untuk 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." (*Sutabri,* 2004: 166).

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

Dapat disimpulkan bahwa diagram konteks adalah suatu diagram sederhana yang meggambarkan suatu proses secara global atau keseluruhan yang didalamnya menjelaskan mengenai seluruh jaringan, masukan dan keluaran.

3.2.3.3.3 *Data Flow Diagram* (DFD)

Pengertian DFD menurut Jogiyanto (2005:701) dalam bukunya yang berjudul *Analisis dan Desain* adalah sebagai berikut:

> "DFD merupakan gambaran suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir, dan akan disimpan. Karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. Maka DFD juga merupakan dokumentasi dari sistem yang baik. (Jogiyanto, 2005:701).

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 (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. (Jogiyanto, 2005 : 700).

3.2.3.3.4 Kamus Data

Definisi mengenai kamus data dijelaskan oleh jogiyanto dalam buku yang berjudul *Analisis dan Desain*. Kamus Data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi." (Jogiyanto, 2005:725).

Definisi lain di jelaskan oleh Al bahra dalam buku yang berjudul *Analisis dan Desain Sistem Informasi*, adalah sebagai berikut:

> "Kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai penganalisis sistem mempunyai dasar pengertian yang sama tentang, masukan, keluaran, penyimpanan, dan proses."(*Al Bahra***,** 2005:70).

Dapat ditarik kesimpulan bahwa kamus data adalah daftar kumpulan elemen-elemen yang tersusun dan saling berhubungan dengan sistem yang didefinisikan secara detail dan tepat. Kamus data dibagi dua yaitu sebagai berikut:

1. Kamus Data Elementer

Kamus data elementer adalah daftar semua data elementer dalam sistem dan disusun berdasarkan abjad. Isi data elementer meliputi nama aribut, *type*, *length*, dan *constraint*.

2. Kamus Data Komposit.

Kamus data komposit adalah semua data bentukan yang terdiri dari dua atau lebih data elementer.

Pada pembuatan kamus data harus terdapat hal-hal berikut ini:

1. Nama Arus Data.

   Nama arus data dicatat pada kamus data, sehingga mereka yang membaca Data Flow Diagram (DFD) memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu.

2. Bentuk Data

   Bentuk data dipergunakan untuk mengelompokan kamus data ke dalam kegunaannya sewaktu perancangan sistem.

3. Arus Data

   Arus data menunjukan dari mana mengakir dan kemana data menuju.

4. Penjelasan

   Penjelasan mengenai makna dari arus data yang dicatat di kamus data.

5. Periode.

   Periode adalah suatu penjelasan kapan waktu untuk memakai data tersebut.

6. Volume.

   Kapasitas waktu data tersebut dipakai.

7. Struktur Data.

   Attribut-attribut data apa saja yang ada dalam data yang dipakai.

3.2.3.3.5 Perancangan Basis Data

a. Normalisasi

Definisi normalisasi dijelaskan oleh jogiyanto dalam buku yang berjudul *Analisis dan Desain*. Normalisasi adalah proses untuk mengorganisasikan file untuk menghilangkan grup elemen yang berulang-ulang." (Jogiyanto, 2005:403).

Definisi lain diutarakan oleh Al Bahra dalam buku yang berjudul *Analisis dan Desain Sistem Informasi*. Normalisasi adalah suatu proses memperbaiki atau membangun dengan model data relasional, dan secara umum lebih tepat dikoneksikan dengan model data logika. (Al Bahra, 2005:169).

Berdasarkan definisi di atas, penulis dapat menyimpulkan bahwa normalisasi adalah suatu proses untuk mengorganisirkan file untuk memperbaiki data relasional agar dapat menghilangkan group elemen yang berulang-ulang sehingga tidak ada lagi elemen yang berulang.

Langkah-langkah pembentukan normalisasi menurut Al Bahra dalam buku yang berjudul *Analisis dan Desain Sistem Informasi*, yaitu sebagai berikut:

A. Bentuk tidak normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan saat menginput.

B. Bentuk normal ke satu (*First Normal Form/1 NF*)

Pada tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi diantara setiap baris pada suatu tabel, dan setiap atribut harus mempunyai nilai data yang *atomatic*.

C. Bentuk normal ke dua (*Second Normal Form/2 NF*)*Third Normal Form*/3 NF)

Walaupun relasi 2-NF memiliki redudansi yang lebih sedikit daripada relasi 1-NF, namun relasi tersebut masih mungkin mengalami kendala bila terjadi *anomaly* peremajaan (*update*) terhadap relasi tersebut.

D. *Boyce-Codd Normal Form* (BCNF)

Boyce-Codd Normal Form (BCNF) didasari pada beberapa ketergantungan fungsional (functional dependencies) dalam suatu relasi yang melibatkan seluruh *candidate key* di dalam relasi tersebut." (Al Bahra**,** 2005:176-188).

b. Tabel Relasi

Pengertian Tabel Relasi menurut Fathansyah (2002:23) adalah Data yang menggambarkan hubungan antara tabel yang satu dengan tabel yang lainnya.

Dari pendapat diatas dapat ditarik kesimpulan tabel relasi adalah sebuah tabel yang menggambarkan hubungan antara tabel yang satu dan yang lainnya karena adanya saling berkaitan diantara tabel tersebut.

c. *Entity Relationship Diagram* (ERD)

Pengertian *Entity Relationship Diagram* (ERD) menurut Al Bahra dalam buku yang berjudul *Analisis dan Desain Sistem Informasi* menjelaskan bahwa: "*Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak."(Al Bahra, 2005:142).

Pendapat lain mengenai pengertian ERD dielaskan oleh Fatansyah dalam buku yang berjudul *Basis Data Lanjut*, menyebutkan bahwa:

> *"Entity Relationship Diagram* yaitu berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari dunia nyata." (Fatansyah, 2004:79).

Elemen-elemen diagram hubungan entitas menurut Al Bahra dalam bukunya yang berjudul *Analisis dan Desain Sistem Informasi*, menjelaskan bahwa:

a. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).
b. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi dengan nama kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya.

c. *Relationsheep Degree* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*." (*Al Bahra***,** 2005:143).

Di dalam sebuah ERD ada yang dinamakan kardinalitas relasi untuk menunjukan jumlah suatu entitas berelasi. Kardinalitas relasi menurut Fathansyah dalm buku yang berjudul *Basis Data Lanjut* adalah Derajat relasi atau kardinalitas menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. (Fatansyah, 2004:77).

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu terdapat 3 macam yaitu:

A. Relasi Satu Ke Satu

Tingkat hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

B. Relasi Satu Ke Banyak

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

C. Relasi Banyak Ke Banyak

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

3.2.4 Pengujian Software

Pengujian *software* adalah proses untuk memastikan apakah semua fungsi sistem bekerja dengan baik, dan mencari apakah masih ada kesalahan pada sistem. Pengujian atau testing software sangat penting untuk dilakukan. Pengujian ini bertujuan untuk menjamin kualitas software, dan juga menjadi peninjauan terakhir terhadap spesifikasi, disain dan pengkodean.

Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah:

1. Pengujian adalah proses *eksekusi* suatu program dengan maksud menemukan kesalahan.
2. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Terdapat 2 metode pengujian *software* yaitu metode *whitebox testing* dan *blackbox testing*. Black box (functionality) testing adalah mengidentifikasi kesalahan yang berhubungan dengan kesalahan fungsional perangkat lunak yang tampak dalam kesalah output. Sedangkan White box (structural) testing/*glass box* testing adalah memeriksa kalkulasi *internal path* untuk mengidentifikasi kesalahan.

Tujuan dilakukannya sebuah pengujian sistem atau software yaitu sebagai berikut:

1. Tujuan Langsung
   a. Identifikasi dan menemukan beberapa kesalahan yang memungkinkan dalam perangkat lunak yang diuji setelah perangkat lunak di perbaiki, di indetifikasi lagi kesalahan dan di tes ualang untuk menjamin kualitas level penerimaan.
   b. Membentuk tes yang efisien dan efektif dengan angggaran dan jadwal yang terbatas.
2. Tujuan Tidak Langsung

Tujuan tidak langsung pengujian perangkat lunak yaitu mengumpulkan daftar kesalahan untuk digunakan daftar pencegahan kesalahan (tindakan *corrective* dan *preventive*)

3.2.4.1 *Black-box Testing*

Dalam skripsi ini penulis menggunakan metode *blackbox testing* untuk melakukan pengujian. Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada output yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi. *Black box* juga didefinisikan sebagai pengujian yang memegang perhitungan mekanisme internal sistem atau komponen dengan kebutuhan fungsional tertentu.

Test case ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutakhirannya.

Uji coba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan *performa*.
5. Kesalahan inisialisasi dan terminasi.

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 Testing*

Equivalence partitioning adalah metode pengujian black-box yg memecah atau membagi domain input dari program ke dalam kelas-kelas data sehingga test case dapat digunakan untuk menggeneresi kasus uji. Perancangan test case equivalence partitioning berdasarkan evaluasi kelas equivalence untuk kondisi input yg menggambarkan kumpulan keadaan yg valid atau tidak. Kondisi input dapat berupa nilai numeric, range nilai, kumpulan nilai yg berhubungan atau kondisi Boolean. Bertujuan untuk menemukan kelas-kelas kesalahan berdasarkan pada kesamaan kelas-kelas kondisi input. Sebuah kelas yang *ekivalen* mereprensentasikan kumpulan

status/kondisi yng valid atau tidak valid. Sebuah input dapat berupa nilai numerik yang spesifikas, rentang nilai, kumpulan nilai yang berkaitan, atau kondisi boolean. Kelas ekivalen dapat didefinisikan dengan kondisi berikut:

1. Jika kondisi input mensyaratkan rentang nilai atau nilai yang spesifik, maka sebuah kelas ekivalen yang *valid* dan dua buah kelas ekivalen yang tidak *valid* akan terbentuk.
2. Jika sebuah kondisi input mensyaratkan sebuah boolean atau anggota dari sebuah himpunan, maka sebuah kelas ekivalen yang valid dan sebuah kelas ekivalen yang tidak valid akan terbentuk.
3. Kasus uji untuk setiap domain input *item* data harus dikembangkan dan dieksekusi.

2. *Boundary Value Analysis*

Boundary Value Analysis digunakan pada saat banyaknya kesalahan yang terjadi pada kesalahan masukan. *Boundary Value Analysis* mengijinkan untuk menyeleksi kasus uji yang menguji batasan niali input. *Boundary Value Analysis* merupakan komplemen dari *equivalence portitioning*. Lebih pada memilih elemen-elemen di dalam kelas *ekivalen* pada bagian sisi batas dari kelas.

3. *Comparison*

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. *Sampel*

Sampel melibatkan beberapa nilai yang terpilih dari sebuah kelas *ekivalen* dan mengintegrasikan nilai pada kasus uji.

5. *Behavior*

Masukkan pada *software* atau sistem terbagi menjadi beberapa kelompok yang diharapkan akan menghsilkan *behavior* yang mirip, sehingga biasanya masukan ini diproses dengan cara yang sama. *Behavior testing* adalah dengan cara melakukan pengecakan sacara berulang-ulang. Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali. misalnya pada pengujian struktur data *stack*.

6. *Robusness*

Data input dipilih diluar spesifikasi yang telah didefinisikan. Maksud dari definisi tersebut adalah mencoba testing dengan cara memasukan data yang tidak sesuai dengan apa yang seharusnya dimasukan atau bukan yang tidak sesuai pendeklarasian variable. Tujuan dari pengujian ini adalah membuktikan bahwa

tidak ada kesalahan jika data yang dimasukan tidak valid, sehingga akan terlihat kesalahan pada sistem tersebut.

7. *Requipment*

Requirement adalah sebuah kemampuan yang harus dimiliki dari suatu *software.* Kemampuan ini dapat ditujukan untuk memecahkan suatu permasalahan ataupun diperlukan untuk memenuhi ketentuan-ketentuan tertentu (seperti standar tertentu, keputusan manajemen, ataupun alasan-alasan politis).

Requirement dapat dibedakan menjadi tiga jenis, yaitu :

1. *User requirement* (kebutuhan pengguna)

Pernyataan tentang layanan yang disediakan sistem dan tentang batasanbatasan operasionalnya. Pernyataan ini dapat dilengkapi dengan gambar/diagram yang dapat dimengerti dengan mudah.

2. *System requirement* (kebutuhan sistem)

Sekumpulan layanan/kemampuan sistem dan batasan-batasannya yang ditulis secara detil. Sistem *requirement document* sering disebut functional specification (spesifikasi fungsional), harus menjelaskan dengan tepat dan detil. Ini bisa berlaku sebagai kontrak antara klien dan pembangun.

3. *Software design specification* (spesifikasi rancangan PL)

Gambaran abstrak dari rancangan *software* yang menjadi dasar bagi perancangan dan implementasi yang lebih detil.

8. *Performance*

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.

9. *Endurance*

Endurance Testing melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai dengan spesifikasi kebutuhan. Spesifikasi kebutuhan pengujian didefinisikan pada tahap spesifikasi kebutuhan atau desain. Manfaat dari *endurance testing* adalah sebagai berikut:

1. Untuk menguji keakuratan operasi matematika (*floating point, rounding off*, dll).
2. Untuk menguji manajemen sumber daya sistem *(resources)* (pembebasan sumber daya yang tidak benar, dll).
3. input/outputs (jika menggunakan *framework* untuk memvalidasi bagian input dan output).

10. *Causes Effect Graph* (CEG)

Teknik ini merupakan suplemen dari *equivalence testing* dengan menyediakan cara untuk memilih kombinasi data input. Melibatkan kondisi input (*causes*) dan kondisi output (*effect*) untuk mencegah pendefinisian kasus uji yang terlalu banyak.

3.3 *Personal Home Page* (PHP)

3.3.1 Pengertian PHP

Menurut Abdul kadir dalam bukunya yang berjudul Dasar Pemograman WEB Dinamis Menggunakan PHP sebagai berikut:

> "PHP merupakan skrip yang dijalankan di server, dimana kode yang menyusun program tidak perlu diedarkan ke pemakai sehingga kerahasiaan kode dapat dilindungi". (Abdul Kadir, 2002).

Sedangkan menurut dokumen resmi PHP, PHP adalah singkatan dari *Personal Hypertext preprocessor*. Secara khusus PHP dirancang untuk membuat web dinamis, artinya PHP dapat berbentuk suatu tampilan berdasarkan permintaan terkini. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *cold fusion atau Perl*. Ciri-ciri situs yang dinamis adalah dapat berinteraksi dengan pengunjung situs, bisa menampilkan informasi-informasi yang berasal dari database, dan halaman web bisa berubah secara otomatis. Berdasarkan tempat dijalankan perintah-perintah program dalam halaman web, pemrograman web dapat dikatagorikan menjadi dua, yaitu pada *server-side programming* perintah-perintah program dijalankan di *web server*, sedangkan *client-side programming* menjalankan perintah pada *client*, dalam hal ini *web browser*. Aplikasi *web* berjalan pada protokol HTTP (*Hypertext Transfer Protocol)* dan semua protokol yang ada di internet selalu melibatkan *server* dan *client*. Ketika seseorang mengetikkan alamat di *web browser*, maka *browser* akan mengirimkan perintah tersebut ke *web server*, jika yang akan diminta adalah file yang mengandung program *server-side* maka *web server* akan menjalankan terlebih dahulu program tersebut dan mengirimkannya ke *browser*. Jika yang

diminta adalah file maka *web server* akan langsung mengirimkan ke *browser* apa adanya.

3.3.2 Sejarah PHP

PHP pertama kali dibuat oleh Rasmus Lerdroft, seorang programmer C. Semula PHP digunakannya untuk menghitung jumlah pengunjung di dalam webnya. Kemudian ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis.

Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan *script* PERL yang dibuatnya untuk membuat halaman webnya menjadi dinamis. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah dapat mengakses *database* dan dapat terintegrasi dengan HTML. Pada tahun 1998 tepatnya tanggal 6 juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang *softwarenya*.

Versi terbaru, yaitu PHP 4.0 keluar pada tanggal 22 mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya *Zend Engine* yang dibuat oleh Zend Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP *scripting engine* dan yang lainnya *adalah build in* HTTP *session*, tidak lagi menggunakan *library* tambahan seperti pada PHP. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikas-aplikasi yang dijalankan di atas teknologi *web browser*, tetapi prosesnya secara keseluruhan dijalankan oleh *web server*.