BAB II

TEORI PENUNJANG

2.1 Pengadaan Barang Alat Rumah Tangga

Kegiatan pengadaan barang rumah tangga yang berlangsung saat ini harus melalui beberapa rangkaian proses. yaitu menentukan kualifikasi jenis barang alat rumah tangga, mulai dari aspek harga, baik itu produk impor atau lokal. Dalam pelaksanaan pengadaan alat rumah tangga terdapat beberapa langkah yaitu :

1. Penyusunan Daftar Pengadaan Barang

   Agar barang yang dibeli sesuai dengan kebutuhan, maka perlu dilakukan identifikasi kondisi keadaan barang secara rinci sehingga tidak ada barang alat rumah tangga yang tidak terdata dan tidak sesuai dengan kebutuhan pengadaan. Langkah-langkah pendataan barang alat rumah tangga adalah sebagai berikut:

   a) Mengidentifikasi jumlah barang alat rumah tangga di setiap instalasi.
   b) Berdasarkan identifikasi jumlah barang alat rumah tangga, di kelompokkan berdasarkan jenis barang dengan memberikan keterangan kondisi keadaan yaitu rusak ringan, rusak berat, dan baik.

2. Kualifikasi Barang

   Agar barang alat rumah tangga yang dibeli sesuai dengan kebutuhan perlu dilakukan penilaian dengan mempertimbangkan jenis barang, merek dagang, barang impor atau lokal. Daftar pengadaan barang dapat menjadi awal dalam mempertimbangkan kebutuhan pengadaan barang alat rumah tangga. Langkah-langkah yang dilakukan pada penilaian kualifikasi pengadaan barang alat rumah tangga sebagai berikut:

   a) Mengidentifikasi kebutuhan pembelanjaan barang alat rumah tangga.
   b) Mengidentifikasi kondisi keadaan barang alat rumah tangga yang masih bagus atau baik.
   c) Menentukan jenis merek serta spasifikasi harga barang alat rumah tangga yang dibutuhkan untuk melengkapi kekurangan yang ada di setiap instalasi.

3. Pengadaan Barang

Pengadaan barang alat rumah tangga harus sesuai dengan kebutuhan di setiap inslatasi ruang pelayanan, dengan melalui beberapa mekanisme:

a) Pengadaan barang yang rusak berat serta rusak ringan menjadi prioritas pembelian sesuai dengan kebutuhan.
b) Penghitungan pembelian barang dibuat melalui koordinasi dengan tim pengadaan alat rumah tangga, dan melibatkan tiap-tiap instalasi layanan.
c) Spesifikasi serta jumlah persatuan harga barang harus diketahui pada saat pembuatan anggaran pembelanjaan alat rumah tangga.[1].

2.2 Logika Fuzzy

Logika fuzzy merupakan suatu cara untuk memetakan suatu ruang masukan ke dalam suatu ruang keluaran. Dalam teori logika fuzzy dikenal himpunan fuzzy (fuzzy set). Merupakan pengelompokan sesuatu berdasarkan variabel yang dinyatakan dalam fungsi keanggotaan (membership function)[2].

2.2.1 Operator logika fuzzy

1. Operator NOT Operator ini berhubungan dengan operasi komplemen pada himpunan. α–predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_A c(x) = 1 - \mu_A(x) \quad \textbf{(II.1)}$$

2. Operator AND

Operator ini berhubungan dengan operasi intersection (irisan) pada himpunan. α–predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{(A\cap B)}(x) = \min[\mu_A(x), \mu_B(x)] \quad \textbf{(II.2)}$$

3. Operator OR

Operator ini berhubungan dengan operasi union (gabungan) pada himpunan. α-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{(A\cup B)}(x) = \max[\mu_A(x), \mu_B(x)] \quad \textbf{(II.3)}$$

2.1.2 Fuzzy Sugeno

Penalaran dengan metode Sugeno output (konsekuen) sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Salah satu metode FIS yang dapat digunakan untuk pengambilan keputusan adalah metode Sugeno. Untuk memperoleh output dari metode fuzzy Sugeno diperlukan 4 tahap, yaitu :

Pada metode Sugeno, output (konsekuen) sistem tidak berupa himpunan fuzzy melainkan berupa konstanta atau persamaan linier. Metode Sugeno terdiri dari 2 jenis , yaitu :

a) Model Fuzzy Sugeno Orde-Nol

Secara umum bentuk model fuzzy Sugeno Orde-Nol adalah : IF (x1 is A1) • (x2 is A2) • (x3 is A3) • …… • (xN is AN) THEN z=k **(II.4)**

dengan Ai adalah himpunan fuzzy ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

b) Model Fuzzy Sugeno Orde-Satu

Secara umum bentuk model fuzzy Sugeno Orde-Satu adalah : IF (x1 is A1) •…• (xN is AN) THEN z = p1*x1 + … + pN*xN + q **(II.5)**

dengan Ai adalah himpunan fuzzy ke-i sebagai anteseden, dan pi adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen. Apabila komposisi aturan menggunakan metode sugeno, maka deffuzifikasi dilakukan dengan cara mencari nilai rata-ratanya (Kusumadewi, 2004).[6]

2.1.3 Sistem Inferensi Fuzzy

Sistem inferensi fuzzy yaitu sistem komputasi yang bekerja atas dasar prinsip penalaran fuzzy. Sistem inferensi fuzzy digunakan untuk memetakan nilai input menjadi nilai output menggunakan logika fuzzy. Sistem inferensi fuzzy dewasa ini banyak digunakan untuk berbagai macam keperluan, seperti sistem pendukung keputusan, penentuan produksi barang, pengenalan pola.Sistem inferensi fuzzy terdiri dari empat unit, yaitu :

1. Fuzzifikasi adalah proses perubahan variabel numerik menjadi variabel linguistik. Fungsi fuzzifikasi digunakan untuk mengubah nilai tegas, misal a∈B, ke suatu himpunan fuzzy C dengan nilai keanggotaan a. Fuzzifikasi

diharapkan dapat membantu menyederhanakan komputasi yang harus dilakukan oleh sistem tersebut dalam proses inferensinya.

2. Penalaran logika fuzzy adalah suatu cara penarikan kesimpulan berdasarkan seperangkat implikasi fuzzy dan suatu fakta yang diketahui. Salah satu aturan penalaran yang paling sering dipergunakan adalah modus ponen, yang didasarkan pada tautologi:(p ^ (p=>q))=>q. Bentuk penalaran modus ponen adalah sebagai berikut : Premis 1x adalah A, Premis 2. Bila x adalah A, maka y adalah B.Kesimpulan y adalah B. Proses penarikan kesimpulan diatas terdiri dari sebuah proposisi tunggal sebagai fakta yang diketahui (premis 1), sebuah proposisi majemuk berbentuk implikasi, yang merupakan suatu kaidah atau aturanyang berlaku (premis 2) dan kesimpulan yang ditarik berdasarkan kedua proposisi tersebut.
3. Basis pengetahuan memuat fungsi-fungsi keanggotaan dari himpunan-himpunan fuzzy yang terkait dengan nilai dari variabel-variabel linguistik yang dipakai dan aturan-aturan berupa implikasi fuzzy.

Defuzzifikasi digunakan menerjemahkan himpunan nilai keluaran kedalam nilai yang tegas. Ada beberapa metode defuzzifikasi dalam pemodelan system fuzzy yaitu :

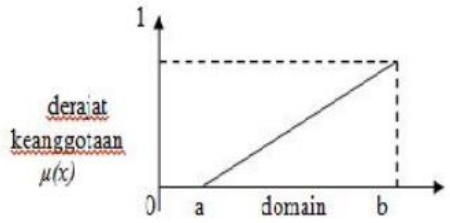
a) Metode Centroid. Pada metode ini, solusi nilai tegas diperoleh dengan mengambil titik pusat (z*) daerah fuzzy.
b) Metode Bisektor. Pada metode ini, solusi nilai tegas diperoleh dengan cara mengambil nilai pada domain fuzzy yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah fuzzy.
c) Metode Mean of Maximum (MOM). Pada metode ini solusi nilai tegas diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.
d) Metode Largest of Maximum (LOM). Pada metode ini, solusi nilai tegas diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e) Metode Smallestof Maximum(SOM). Pada metode ini, solusi nilai tegas diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai kanggotaan maksimum.[7]

2.1.4 Fungsi Keanggotaan

Fungsi keanggotaan *(membership function)* adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai. Berikut representasi fungsi keanggotaan :

1. Representasi Linier Naik

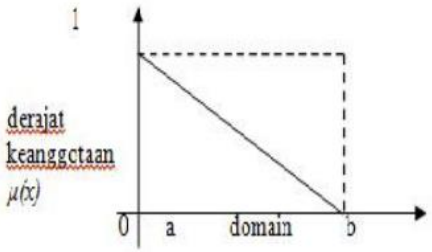


Gambar 2.1 Representasi Linier Naik

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \le a \\ (x-a)/(b-a); & a \le x \le b \\ 1; & x \ge b \end{cases} \quad \textbf{(II.6)}$$

2. Representasi Linier Turun

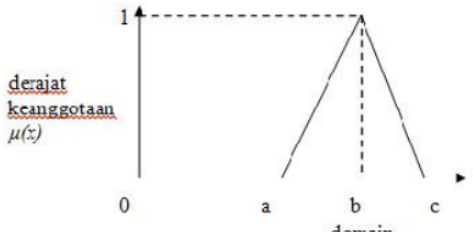


Gambar 2.2 Repesentasi Linier Turun

Fungsi Keanggotaan

$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \le x \le b \\ 0; & x \ge b \end{cases} \quad \textbf{(II.7)}$$

3. Representasi Kurva Segitiga

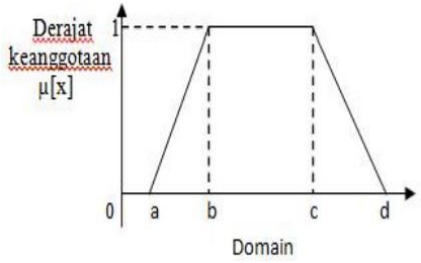


Gambar 2.3 Representasi Kurva Segitiga

Fungsi Keanggotaan

$$\mu[x] = \begin{cases} 0; & x \le a \text{ atau } x \ge c \\ (x-a)/(b-a); & a \le x \le b \\ (b-x)/(c-b); & b \le x \le c \end{cases} \quad \textbf{(II.8)}$$

4. Representasi Kurva Trapesium



Gambar 2.4 Representasi Kurva Trapesium

Fungsi Keanggotaan

$$\mu[x] = \begin{cases} 0; & x \le a \text{ atau } x \ge d \\ (x-a)/(b-a); & a \le x \le b \\ 1; & b \le x \le c \\ (d-x)/(d-c); & x \ge d \end{cases} \quad \textbf{(II.9)}$$

2.2 MAPE (Mean Absolute Percentage Error)

Mean Absolute Percentage Error (MAPE) merupakan suatu ukuran akurasi peramalan dari suatu metode peramalan. Caranya yaitu dengan menghitung selisih dari output yang diperoleh dengan data sebenarnya, kemudian dibagi dengan data sebenarnya. Hasilnya yang berbentuk persentase kemudian dimutlakkan. Perhitungan ini dilakukan pada setiap amatannya, kemudian dirata-ratakan. Hasil peramalan sangat bagus jika nilai MAPE kurang dari 10% sedangkan nilai MAPE dikatakan bagus jika kurang dari 20% (Harun,1999)[8]. MAPE didefinisikan sebagai berikut :

$$MAPE = \frac{\sum_{i=1}^{n} \left| \frac{X_i - F_i}{X_i} \right|}{n} \times 100\% \quad \textbf{(II.10)}$$

Dengan Xi = nilai data asli amatan ke – i

Fi = nilai ramalan amatan ke – i

n = banyak data

2.3 Website

Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. Website merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara animasi sehingga lebih merupakan media informasi yang menarik untuk dikunjungi[4].

Secara terminologi, website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web (WWW)* di internet. Sebuah halaman web adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP (Hypertext Transfer Protocol), yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari website-website tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.Halaman-halaman dari website bisa diakses melalui sebuah URL (*Unicode Resource Locator*) yang biasa disebut *Homepage*. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun, *hyperlink-hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan.

Penemu website adalah Sir Timothy John Tim Berners-Lee, sedangkan website yang tersambung dengan jaringan, pertamakali muncul pada tahun 1991. Maksud dari Tim ketika membuat website adalah untuk mempermudah tukar menukar dan memperbarui informasi kepada sesama peneliti di tempat dia bekerja. Pada tanggal 30 April 1993, CERN (tempat dimana Tim bekerja) menginformasikan bahwa WWW dapat digunakan secara gratis oleh semua orang.

Sebuah website bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari sebuah organisasi, perusahaan, dan biasanya website itu menujukkan beberapa topik khusus, atau kepentingan tertentu. Sebuah website bisa berisi *hyperlink* (pranala) yang menghubungkan ke website lain, jadi, kadangkala perbedaan antara website yang dibuat oleh individu perseorangan dengan website yang dibuat oleh organisasi bisnis bisa saja tidak kentara.

Website ditulis atau secara dinamik di konversi menjadi HTML dan diakses melalui sebuah program software yang biasa disebut dengan *web browser*, yang dikenal juga dengan HTTP *Client*. Halaman web dapat dilihat atau diakses melalui jaringan komputer dan internet, perangkatnya bisa saja berupa komputer pribadi, laptop, ataupun telepon selular.

2.4 UML

UML (Unified Modeling Language) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak[3].

2.4.1 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemrograman berorientasi objek selanjutnya.

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Secara aktual, penggunaan bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian di saat itu. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sally Shaler dan Stephen Mellor (Shaler-

Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Grady Booch (Booch, 1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha-Premerlani-Eddy-Lorensen, 1991), dan masih banyak lagi.

Buku terkenal yang juga berkembang selanjutnya adalah karangan Ivar Jacobson (Jacobson, 1992) yang menerangkan perbedaan pendekatan yang fokus pada *use case* dan proses pengembangan. Sekitar lima tahun kemudian muncul buku yang membahas mengenai metodologi berorientasi objek yang diikuti dengan buku-buku yang lainnya. Didalamnya juga membahas mengenai konsep, definisi, notasi, terminologi, dan proses mengenai metodologi berorientasi objek.

Karena banyaknya metodologi-metodologi yang berkembang pesat itu, maka muncul ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modeling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainya dimana James R. Rumbaigh, Grandy Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language* (UML).

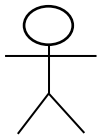
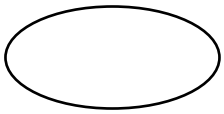
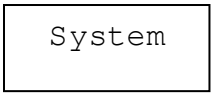
2.4.2 Diagram UML

Berikut akan dijelaskan diagram yang akan digunakan dalam pembangunan sistem monitoring distribusi bantuan berbasis website ini, yaitu *use case diagram*, *class diagram, sequence diagram*, dan a*ctivity diagram*.

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk (*behavior)* aplikasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan aplikasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah aplikasi dan siapa saja yang berhak mengguankan fungsi-fungsi itu. Penjelasan tentang simbo-simbol *usecase*dapat dilihat pada tabel 2.1.

Tabel 2.1 Definisi Simbol Use Case Diagram

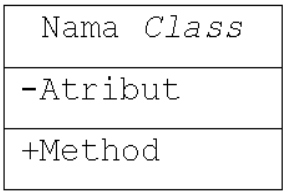
Simbol	Nama Simbol	Fungsi
	Aktor	Merupakan orang, proses, atau sistem lain yang berinteraksi dengan aplikasi yang akan dibuat di luar aplikasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang
	*Use Case*	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
	Association	Merelasikan Aktor dengan *use case*
	*System Boundary*	Menggambarkan batasan sistem terhadap lingkungannya

2. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut, merupakan variable-variabel yang dimiliki oleh satu kelas.
2. Operasi, atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Class diagram dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sesuai. Contoh *class diagram* dapat dilihat pada gambar 2.5.

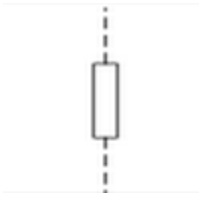
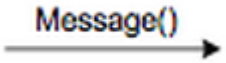


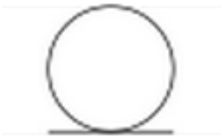
Gambar 2.5 Contoh Class Diagram

3. *Sequence Diagram*

Sequence diagram menggambarkan *behavior* objek pada *use case* dengan mendeskripsikan waktu hidup dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan melihat scenario yang ada pada use case. Penjelasan tentang simbol-simbol *sequence diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 Definisi Simbol Sequence Diagram

Simbol	Nama Simbol	Fungsi
	*Lifelines*	Menggambarkan waktu mulai dan berakhirnya sebuah message
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktfitas yang terjadi
	*Actor*	Digunakan untuk menggambarkan user / pengguna

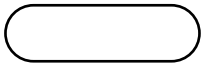
	*Boundary*	Digunakan untuk menggambarkan sebuah form atau antarmuka
	*Control Class*	Digunakan untuk menghubungkan boundary dengan entity
	*Entity Class*	Menyimpan data/informasi yang digunakan dalam perangkat lunak

4. *Activity Diagram*

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah *activity diagram*menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Tabel 2.3 adalah penjelasan tentang symbol-simbol *activity diagram*.

Tabel 2.3 Definisi Simbol Activity Diagram

Simbol	Nama Simbol	Fungsi
	*Initial State*	Pernyataan awal dari sebuah aktivitas
	*Final State*	Pernyataan akhir dari sebuah aktivitas
	*Transisi*	Menggambarkan alur dari aksi – aksi yang ada pada aktivitas

	Aksi	Menggambarkan aksi yang ada pada aktivitas

2.5 PHP

PHP (*Hypertext Preprocessor)* yaitu bahasa pemrograman web server-side yang bersifat *open source*. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (*server side* HTML *embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua script PHP dieksekusi pada server dimana *script* tersebut dijalankan.

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 *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram 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 diubah 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[5].