

**APLIKASI WEB USULAN PENETAPAN ANGKA KREDIT  
JABATAN FUNGSIONAL DOSEN**

Tesis  
Untuk memenuhi sebagian persyaratan  
Mencapai derajat Sarjana S-2  
Program Studi Magister Sistem Informasi



Oleh:  
Teguh Yuwono  
J4F008033

PROGRAM PASCASARJANA  
UNIVERSITAS DIPONEGORO  
SEMARANG  
2011

## BAB 1

### PENDAHULUAN

#### 1.1. Latar Belakang

Dengan perkembangan teknologi layanan komputasi, *Web Applications Development* (WAD) menggantikannya dengan sumber daya untuk menggunakan kembali layanan yang tersedia di Internet. Dengan demikian, perancangan dan pemrograman fase *account* tidak akan lagi untuk rasio yang besar dalam siklus pengembangan perangkat lunak. Sebaliknya, persyaratan rekayasa (ketepatan, reliabilitas, kegunaan, aksesibilitas, kinerja dan kesesuaian), sebuah fase yang terjadi sebelum desain dan pemrograman, akan memainkan peran yang lebih penting yang menentukan keberhasilan WAD. Zhang (2008)

Dalam kenyataannya rekayasa harus dipikirkan dengan baik sebagai desain dan pemrograman, namun kekurangan yang telah menyebabkan banyak proyek dengan sedikit persyaratan dan menyalahkan sebagai alasan utama bagi banyak kegagalan perangkat lunak. Oleh karena itu, persyaratan teknik sekarang bergerak ke permukaan memperoleh peningkatan signifikan dalam rekayasa perangkat lunak untuk layanan aplikasi berorientasi *web*. Zhang (2008).

Aplikasi *web* memiliki persyaratan karakteristik yang baru menyebabkan mereka untuk mengubah lebih cepat. Hal ini membuat pemodelan tradisional dan metode validasi persyaratan cukup untuk memberikan dukungan yang memadai untuk aplikasi *web*. Persyaratan aplikasi *web* tidak hanya berasal dari analisis domain umum dan personalisasi persyaratan pengguna beragam, tetapi juga ketersediaan layanan *web* terkait. Persyaratan aplikasi *web* juga berkembang saat

*web* digunakan, sehingga aplikasi *web* harus memenuhi untuk mengeksplorasi pendekatan rekayasa, mengidentifikasi, menjelaskan, memvalidasi dan persyaratan mengelola aplikasi *web*. Zhang (2008).

Penggunaan aplikasi *web* untuk menyelesaikan tugas sehari-hari dalam perusahaan yang mengarah ke aplikasi *web* dengan lebih kompleks. Beberapa aplikasi *web* menimbulkan kompleksitas lebih lanjut. Para pengguna memanfaatkan aplikasi *web* untuk menyelesaikan tugas-tugas bisnis tertentu dengan cara yang paling efisien. Akibatnya, kemungkinan terlalu banyak navigasi dalam sistem dan tugas-tugas bisnis yang tersebar dalam satu jaringan oleh penggunaan aplikasi *web* dengan menanggung risiko pengguna akhir yang menggunakan sistem tersebut. *Sonneck* (2003).

Pembelajaran secara tradisional dan pembelajaran secara *online* dibahas dalam tesis *Hughes* (2009). Pada kajian tersebut disarankan tentang informasi mengenai beragam budaya *online* diperguruan tinggi.

Dosen dalam pengumpulan angka kredit dosen masih sering mengalami keterlambatan dan masih harus diingatkan oleh pimpinan fakultas. Untuk mengingatkan seorang dosen yang lama tidak naik jabatan, petugas kepegawaian harus mencari file-file setiap dosen mana yang harus diberi peringatan tersebut, ini akan memerlukan tenaga dan waktu.

Penilaian angka kredit adalah suatu proses penilaian prestasi dosen dilihat dari aktivitas-aktivitas yang telah dilakukan dosen berdasarkan pada pedoman penilaian angka kredit jabatan fungsional dosen. Dosen tidak melakukan pengajuan

yang teratur dan disiplin. Aktivitas dosen juga tidak terarah dalam pemenuhan unsur-unsur yang menjadi syarat kenaikan jabatan fungsional. Pengumpulan, pencatatan, penghitungan, dan penyusunan dokumen pengajuan masih dilakukan secara manual sehingga lambatnya pengajuan yang dilakukan dosen. Monitoring aktivitas dan nilai yang diperoleh tidak bisa diketahui secara langsung. Saptadi, S. (2008).

Masalah keterlambatan dosen dalam naik jabatan fungsional dan kesulitan mencari file-file dosen yang terlambat naik jabatan dan agar dapat dimonitor, perlu dibuatkan sistem informasi dengan komputer sebagai alat bantu yang baik dalam pengumpulan data, pencatatan, penghitungan, dan penghasil informasi yang cepat dan akurat. Sistem Informasi Penetapan Angka Kredit (SIPAK) Dosen menggunakan *web applications* dirancang untuk membantu mempermudah dalam pelaksanaan pengisian penetapan angka kredit. SIPAK diisikan dalam web oleh Dosen yang bersangkutan sebagai syarat pengajuan jabatan fungsional ke jenjang fungsional yang lebih tinggi.

Dari berbagai pembahasan diatas, di perguruan tinggi yang berhubungan dengan dosen khususnya mengenai isian angka kredit jabatan fungsional dosen dari layanan manual dapat diubah menjadi layanan yang lebih efektif dan efisien melalui internet dengan memanfaatkan aplikasi web (*web applications*), untuk itu perlu adanya “aplikasi web usulan angka kredit jabatan fungsional dosen.”

## 1.2. Perumusan Masalah

Membangun desain aplikasi web untuk usulan angka kredit jabatan fungsional dosen.

## 1.3. Batasan Masalah

Desain atau kerangka bentuk pembuatan aplikasi web.

- 1) Rancang bangun dibatasi hanya pada aplikasi web usulan penetapan angka kredit jabatan fungsional dosen untuk syarat kenaikan jabatan, tidak membahas prosedur kenaikan jabatannya, karena prosedur kenaikan jabatan fungsional tergantung pribadi dosen yang ingin terus mengurus kenaikan jabatan fungsionalnya yang melibatkan tim penilaian angka kredit.
- 2) Aplikasi oleh user hanya dapat dilakukan oleh *client* yang telah terdaftar dalam *database server*.
- 3) Jaringan untuk komunikasi antara *server* dan *client* menggunakan intranet atau internet.

## 1.4. Keaslian Penelitian

Natakusumah (2008) membahas penelitian merancang sistem informasi Penetapan Angka Kredit (PAK) untuk jabatan peneliti **yang hanya** sebatas berbasis komputer, sehingga dapat memudahkan dalam membuat laporan penentuan angka kredit, memudahkan pemasukan dan pencarian data sehingga dapat menghasilkan informasi yang akurat, tepat waktu dan relevan.

Karnila (2007), juga merancang sistem informasi Penetapan Angka Kredit (PAK) fungsional berdasarkan **sebatas** sistem komputerisasi. Sistem ini menghasilkan laporan PAK, dan memiliki fasilitas untuk *entry* data dan pencarian, sehingga menghasilkan informasi yang memenuhi kriteria akurasi, ketepatan waktu, dan relevansi.

Achlison (2005) menulis tentang mempermudah penukaran informasi akademik dari kampus yang berjauhan perlu suatu sistem informasi akademik yang dapat diakses menggunakan jaringan WAP dan GPRS. Pembahasan Achlison ini mengenai Kartu Hasil Studi (KHS) lewat telepon seluler.

Sistem Informasi Penetapan Angka Kredit Dosen (SIPAK Dosen) jabatan fungsional berdasarkan sistem komputerisasi ditambah *web applications* merupakan fasilitas komputer yang berbeda dari yang telah ada saat ini. Karena untuk meng-*entry* data angka kredit jabatan fungsional menggunakan komputer merupakan hal yang dianggap biasa. Penggunaan jaringan Internet saat ini juga merupakan hal umum.

Telah adanya Penetapan Angka Kredit berbasis komputer, menurut Natakusumah dan Karlina, untuk itu perlu dibuatkan aplikasi web sebagai nilai tambah, sehingga pengisian angka kredit jabatan fungsional menggunakan aplikasi web lewat jaringan Internet merupakan nilai tambah yang sangat berbeda dengan sistem yang telah ada, jadi ini merupakan SIPAK yang baru.

### **1.5. Manfaat Hasil Penelitian**

Secara *online*, dosen dapat mengaplikasikan web dalam mengisikan angka kreditnya yang akan digunakan sebagai syarat-syarat pemenuhan penetapan angka kredit jabatan fungsional, juga memudahkan petugas kepegawaian dalam memeriksa data dosen dengan perolehan angka kreditnya.

### **1.6. Tujuan Penelitian**

- 1) Tujuan penelitian ini membuat rancang bangun *web applications* dengan kerangka bentuk semantik yang cocok untuk menemukan *web* sesuai permintaan pengguna. Untuk itu diperlukan suatu komposisi yang saling terkait beberapa aplikasi web, tujuannya adalah untuk memperoleh hasil terbaik bagi pengguna.
- 3) Mengintegrasikan beberapa web yang dibuat menggunakan mesin rekomendasi yang merupakan bagian integral dari fase integrasi sistem membuat masukan angka kredit menjadi rekomendasi final.

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Sebuah aplikasi *web* adalah setiap program yang berjalan sepenuhnya dari dalam *browser*, misalnya di *Microsoft Internet Explorer (rossiterandco.com)*.

Dalam rekayasa perangkat lunak, sebuah aplikasi *web* atau *webapp* adalah sebuah aplikasi yang diakses melalui *web browser* melalui jaringan seperti Internet atau intranet. Sebuah aplikasi *web* adalah aplikasi yang menggunakan *web browser* sebagai klien. Aplikasi ini dapat sederhana seperti papan pesan atau tamu *sign-in* buku di situs web, atau sebagai kompleks pengolah kata atau *spreadsheet*.

Sebagai aplikasi berbasis di WWW semakin bertambah baik dari segi skala dan kompleksitas, telah dikenal bahwa WWW telah berkembang dari sebuah media informasi *hypermedia* menjadi *platform* aplikasi baru didistribusikan, dan oleh karena itu tepat dan prospektif untuk melihat aplikasi *web* sebagai perangkat lunak. Mengembangkan dan memelihara web adalah kompleks dituntut proses yang sistematis dan metodologi teknik yang efektif. Sebuah model yang sesuai dari aplikasi *web* yang dapat menangkap fitur-fiturnya sangat penting dan mendasar bagi pembentukan pendekatan tersebut. Berdasarkan pemahaman tentang aplikasi *web* sebagai perangkat lunak, sebuah aplikasi *web* komponen berorientasi model di mana aplikasi *web* dianggap sebagai kumpulan komponen, masing-masing memiliki fungsi sendiri dan bekerja sama dengan orang lain melalui *interface*. Zhao (1999).

Proposisi nilai dari sebuah kerangka aplikasi *web* disajikan untuk menggambarkan bagaimana kerangka kerja dapat meningkatkan produktivitas pengembangan aplikasi dan kualitas. Sebuah skema taksonomi komprehensif didefinisikan untuk mengklasifikasikan berbagai kerangka kerja dan kerangka kerja perangkat lunak aplikasi *web* ke dalam kategori yang sesuai. Kartu referensi dibangun untuk merangkum aspek-aspek kunci dari kerangka kerja aplikasi *web*. Tony, C. (2006).

Situasi saat ini dalam pengembangan aplikasi *web* adalah mengingatkan pada masa-masa awal sistem perangkat lunak, ketika kualitas benar-benar tergantung pada kemampuan individu. Bahkan, aplikasi web biasanya dikembangkan tanpa mengikuti proses formal model, persyaratan tidak ditangkap dan desain tidak dianggap, pengembang cepat pindah ke tahap implementasi dan memberikan aplikasi tanpa mengujinya. Tidak berbeda dari sistem perangkat lunak yang lebih tradisional, namun kualitas aplikasi *web* adalah atribut, kompleks multidimensi yang melibatkan beberapa aspek, termasuk ketepatan, reliabilitas, kemampuan pemeliharaan, kegunaan, aksesibilitas, kinerja dan kesesuaian dengan standar. Ricca, F. (2004).

Menurut *Jaffe, J. Chief Executive Officer W3C (2010)*. Teknologi web terus menjadi kendaraan bagi setiap industri untuk menggabungkan perubahan pesat menjadi cara mereka melakukan bisnis. Peningkatan inovasi, adalah tempat di mana industri datang bersama untuk menetapkan standar untuk web secara terbuka dan kolaboratif mode. W3C adalah berfokus pada teknologi untuk mengaktifkan

akses web di mana saja, kapan saja, dengan menggunakan perangkat apapun. Hal ini termasuk akses web dari ponsel dan perangkat selular lainnya.

Ada tiga *High Availability* (HA) faktor dalam web: ketersediaan, mencapai-kemampuan, dan kinerja. Dalam sebuah aplikasi HA, semua layanan sistem penting dan dapat beropansi dengan benar. Semua layanan mampu yang berarti dapat dipanggil melalui jaringan, dan memenuhi standar minimum: jaringan, aplikasi, dan sumber daya server. (*Kneiling, J. 2004*)

Sebuah aplikasi web adalah sebuah aplikasi yang dapat diakses melalui jaringan Internet atau intranet. Istilah ini juga dapat berarti sebuah aplikasi perangkat lunak komputer yang di-*host* dalam lingkungan yang dikendalikan *browser* atau dikodekan dalam bahasa *markup browser* seperti HTML dan bergantung pada *browser* umum untuk membuat aplikasi dieksekusi. (*Berkovich, S. 2010*).

Aplikasi biasanya dibagi menjadi potongan logis yang disebut ‘tingkatan’, dimana setiap lapis diberi peran aplikasi tersendiri, namun aplikasi web memungkinkan n-berjenjang dengan adanya pengembangan *web*, meskipun banyak variasi, ada tiga tingkatan yang paling umum: presentasi, aplikasi, dan penyimpanan. (*Berkovich, S. 2010*)

## **2.2. Landasan Teori**

### **2.2.1 Sejarah Web Aplikasi (*Web Applications*)**

Dalam wikipedia di dijelaskan sejarah *web applications* (alikasi web) bahwa setiap aplikasi punya klien sendiri, program yang berfungsi sebagai *user interface* dan harus diinstal secara terpisah pada komputer pribadi masing-masing pengguna. *Upgrade* ke bagian *server* dari aplikasi biasanya akan membutuhkan *upgrade* ke *klien* diinstal pada setiap *workstation* pengguna, menambah biaya dan produktivitas menurun.

Sebaliknya, aplikasi web menggunakan dokumen *web* yang ditulis dalam format standar seperti HTML (dan XHTML), yang didukung oleh berbagai *browser web*. Umumnya, setiap halaman *web* individu diserahkan kepada pengguna sebagai dokumen statis, tetapi urutan halaman dapat memberikan pengalaman interaktif, sebagai masukan pengguna dikembalikan melalui elemen formulir *web* tertanam di halaman *markup*. Selama sesi berlangsung, *web browser* menafsirkan dan menampilkan halaman, dan bertindak sebagai *klien* universal untuk setiap aplikasi web.

Pada tahun 1995, *Netscape* memperkenalkan sebuah bahasa *client-side scripting* yang disebut *JavaScript*, yang memungkinkan programmer untuk menambahkan beberapa elemen yang dinamis untuk antarmuka pengguna. Sampai saat itu, semua data harus dikirim ke *server* untuk diproses, dan hasilnya disampaikan melalui halaman HTML statis dikirim kembali ke pengguna.

Pada tahun 1996, *Macromedia Flash* diperkenalkan, sebuah animasi vektor yang dapat ditambahkan ke *browser* sebagai *plug-in* untuk animasi menanamkan pada halaman *web*. Ini memungkinkan penggunaan bahasa *scripting* untuk interaksi program pada sisi *klien* tanpa perlu berkomunikasi dengan *server*.

Pada tahun 1999, konsep *web application* diperkenalkan dalam bahasa *Java* dalam versi Spesifikasi *Servlet 2.2*. Pada waktu itu baik *JavaScript* dan *XML* sudah dikembangkan, tapi *Ajax* itu masih belum diciptakan dan objek *XMLHttpRequest* diperkenalkan di *Internet Explorer 5* sebagai objek *ActiveX*.

Pada tahun 2005, *Ajax* diciptakan, dan aplikasi seperti *gmail* mulai membuat sisi pengguna lebih banyak dan lebih interaktif.

Sistem operasi *Webconverger* menyediakan sebuah antarmuka untuk aplikasi *web*. Antarmuka *web* batasan yang sangat sedikit pada fungsi klien. Melalui *Java*, *JavaScript*, *DHTML*, *Flash*, *Silverlight* dan teknologi lainnya, metode aplikasi-spesifik seperti menggambar di layar, bermain audio, dan akses ke *keyboard* dan *mouse* memudahkan pengguna. Banyak pelayanan telah bekerja untuk menggabungkan semua ke dalam *interface* yang lebih akrab yang mengadopsi tampilan dari sistem operasi. Tujuan umum teknik seperti *drag and drop* juga didukung oleh teknologi tersebut. *Web developer* sering menggunakan *script* disisi pengguna untuk menambah fungsionalitas, terutama untuk menciptakan pengalaman interaktif yang tidak memerlukan *reload* halaman. Baru-baru ini, teknologi telah dikembangkan untuk mengkoordinasikan *scripting client-side* dengan teknologi *server-side* seperti *PHP*. *PHP* sebuah teknik *web development* menggunakan kombinasi dari berbagai teknologi, adalah contoh teknologi yang menciptakan pengalaman yang lebih interaktif.

Sebuah halaman *web* adalah dokumen yang dibuat dengan *HTML* yang merupakan bagian dari kelompok dokumen *hypertext* atau sumber daya yang

tersedia di WWW. Secara keseluruhan, dokumen-dokumen dan bentuk sumber daya apa yang dikenal sebagai sebuah situs *web*.

Dalam *techdreams.in*, di jelaskan bahwa web aplikasi bisa menggunakan bahasa dan alat-alat bantu lain, antara lain: *PHP*; *HTML*; *JavaScript*; *CSS*, dan lain sebagainya. *Web* aplikasi biasanya menggunakan kombinasi beberapa *script* (*PHP*, *HTML*, *Javascript*, *CSS*, dll) untuk mengembangkan aplikasi. Transaksi *script* sisi pengguna dengan penyajian informasi sementara transaksi *server-side script* dengan semua hal yang keras seperti menyimpan dan mengambil informasi. Sebuah *web* aplikasi meringankan tanggung jawab bangunan klien untuk jenis tertentu dari komputer atau sistem operasi tertentu. Karena klien berjalan di *web browser*, pengguna bisa menggunakan komputer lewat *Internet Explorer* atau *Firefox*, meskipun beberapa aplikasi memerlukan *browser web* tertentu.

Halaman *web* dapat berisi *link hypertext* ke tempat-tempat lain dalam dokumen yang sama, ke dokumen lain di situs yang sama atau ke dokumen di situs-situs lain. Mereka juga dapat berisi mengisi formulir, foto, gambar, suara, dan video yang dapat diklik untuk *men-download*.

*Web Server* suatu istilah yang sering digunakan untuk menggambarkan sebuah komputer di situs *web*. Sebenarnya istilah ini berarti perangkat lunak yang berjalan pada komputer yang memungkinkan halaman *web* yang akan diminta dan kemudian dikirim ke *web browser* pengguna.

### **2.2.2 Desain Aplikasi web dengan HTML**

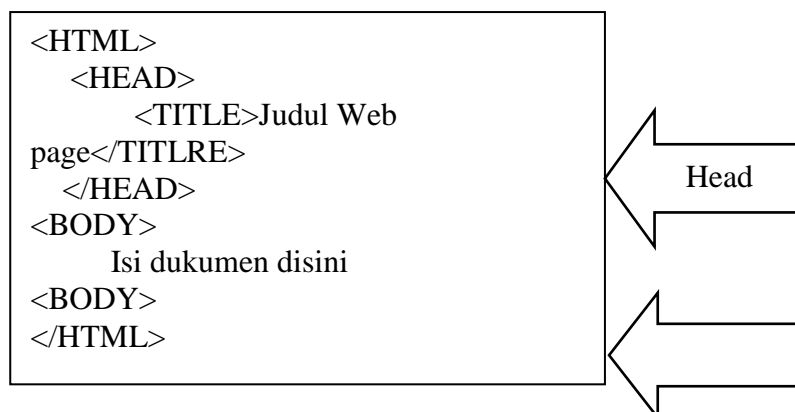
HTML merupakan sebuah bahasa untuk menggambarkan struktur halaman *web*. HTML memberikan cara untuk:

- a) Mempublikasikan dokumen *online* dengan judul, teks, tabel, daftar, foto, dan lain lain.
- b) Ambil informasi *online* melalui *hypertext link*.
- c) Desain formulir untuk melakukan transaksi dengan layanan jarak jauh, untuk digunakan dalam mencari informasi, membuat *reservasi*, memesan produk, dan lain lain.
- d) Serta menyebar lembaran, klip video, klip suara, dan aplikasi lain secara langsung dalam dokumen.

Dengan HTML, dapat menggambarkan struktur halaman menggunakan *markup*. Unsur-unsur bahasa potongan konten label seperti "ayat," "daftar," "tabel," dan seterusnya.

HTML adalah bahasa dasar *web scripting* bersifat *client side*, penulisan HTML dapat dilihat pada gambar 2.1.

(a) Memiliki struktur minimal dari dokumen HTML



## Body

**Gambar 2.1** Struktur Dokumen HTML (Suteja, BR.,2007,h.2)

- (b) Gambar 2.1 menunjukkan TAG pembuka `<TAG>` dan tidak semua tag mempunyai penutup `</TAG>`, tag memberi intruksi kepada *interpreter* kepada *browser* bahwa dokumen tersebut adalah HTML dengan tag `<HTML></HTML>` di awal dan di akhir dokumen. Dalam dokumen HTML penggunaan tag-tag penutup HTML disesuaikan dengan urutan penggunaan tag tersebut.
- (c) Di dalam sebuah tag HTML terdapat atribut tag. Atribut yang terkandung dalam tag satu dengan lainnya belum tentu sama. Atribut yang dipakai tidak memiliki urutan pendefinisian tertentu. Atribut tag digunakan untuk memodifikasi sifat-sifat dari tag, tergantung dari nilai yang diberikan, namun ada juga atribut tag yang tidak mempunyai nilai.
- (d) Penulisan TAG bersifat *non case sensitive*, artinya penulisan huruf kecil atau besar mempunyai nilai sama. Penulisan `<HTML>` sama nilainya dengan penulisan `<html>`.
- (e) Penulisan tanda “<” harus langsung diikuti nama TAG dan ditutup tanda “>” tidak boleh ada spasi.
- (f) Dalam dokumen HTML diperlukan tag `<HEAD> </HEAD>`, dalam tag ini diberikan judul *web page* dengan menggunakan tag `<TITLE>` judul *web page*`</TITLE>`, dan dalam tag `<BODY></BODY>` berisikan isi *web page*.

Untuk ekstensi dari *file* HTML ditambahkan .html atau .htm di belakang nama *file*-nya.

- (g) Penulisan tag-tag HTML tidak terpengaruh langsung oleh perintah *keyboard* sehingga diperbolehkan menyusun tag-tag HTML secara berterusan horizontal, seperti: <HTML><HEAD><TITLE> Judul Web page</TITLE></HEAD> dan seterusnya.

Struktur halaman web menggunakan HTML mempunyai beberapa kelengkapan atribut antara lain: *body*, *heading*, *paragraph*, *breaking now*, *horizobtal ruler*, huruf *bold*, *italic*, *fontasi*, *image*, *hyperlink*, *table*, *password*, *hidden*, *select*, *frame*, *button*, dan lain sebagainya. Atribut-atribut tersebut dapat dijelaskan sebagai berikut:

#### 1) **Body**

Gambar 2.1 mempunyai *body*, atribut tersebut ada yang mempunyai nilai dan ada yang tidak mempunyai nilai. TAG <BODY> memiliki atribut:

```
<BODY bgcolor = "#..."background="...">
```

Warna yang digunakan dalam desain web memiliki berbagai pilihan yang tersedia untuk menentukan elemen dokumen *web*. Warna dapat diperinci sebagai triplet RGB (*Red-Green-Blue*) dalam format heksadesimal maksimal adalah FFF (setara dengan 255 decimal) tiap komposisinya. Bila diinginkan warna merah (*red*) nilainya adalah "#FF0000", warna hijau (*green*) nilainya "00FF00", dan warna biru nilainya "0000FF". atau dapat juga langsung menggunakan *pre-define color* yaitu langsung ditulis *red*, *green*, *blue* dan sebagainya. Untuk atribut *background*

nama *file* gambar lengkap dengan lokasi *folder* dan *file*-nya. Format *file* gambar yang didukung oleh HTML antara lain JPEG, PNG, dan GIF.

HTML 4.01 mempunyai spesifikasi enam belas warna yang dapat lihat pada gambar 2.2.

Color	Hexadecimal	Color	Hexadecimal	Color	Hexadecimal	Color	Hexadecimal
<u>aqua</u>	#00FFFF	<u>green</u>	#008000	<u>navy</u>	#000080	<u>silver</u>	#C0C0C0
<u>black</u>	#000000	<u>grey</u>	#808080	<u>olive</u>	#808000	<u>teal</u>	#008080
<u>blue</u>	#0000FF	<u>lime</u>	#00FF00	<u>purple</u>	#800080	<u>white</u>	#FFFFFF
<u>fuchsia</u>	#FF00FF	<u>maroon</u>	#800000	<u>red</u>	#FF0000	<u>yellow</u>	#FFFF00

**Gambar 2.2** Macam warna HTML

Model kerja HTML diawali dengan permintaan suatu halaman *web* oleh pengguna, user (*browser*). Berdasarkan URL atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *web server* akan mencari berkas yang diminta dan memberikan isinya ke *browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan mengaplikasikannya ke layar pemakai, Kadir (2003. h.4-5).

## 2) *Heading*

*Heading* digunakan untuk menampilkan format huruf besar dan dicetak tebal, biasa digunakan untuk judul atau topik utama, ukuran terbesar <H1> sampai terkecil <H6>. Untuk mengatur posisi *heading* digunakan atribut "align" seperti (H1 align="...."> ditutup dengan </H1> dengan atribut *left* untuk rata kiri, *right* untuk rata kanan, dan *center* untuk posisi tengah.

### **3) Paragraf**

Menformat paragraf rata kiri, rata kanan, posisi tengah, dan *justify* digunakan tag pembuka <P align="...."> dan ditutup </P>. Isi atribut "align" adalah *left* untuk rata kiri, *right* untuk rata kanan, *center* untuk posisi tengah, dan *justify* untuk rata kiri dan kanan. Antara paragraf dalam HTML akan dibatasi secara otomatis oleh sebuah jeda baris.

### **4) Breaking Now**

*Breaking now* digunakan untuk pindah baris baru dengan tag <BR>. Tag yang satu ini salah satu tag HTML yang tidak mempunyai tag penutup.

### **5) Horizontal Ruler**

Untuk membuat garis horizontal digunakan tag <HR> , untuk perataan digunakan atribut "align", untuk mengatur ketebalan baris dalam *pixel* digunakan "size", untuk mengatur lebar baris dalam *pixel* atau persen digunakan "width".

Penggunaan satuan persen memiliki arti bahwa ukuran baris diambil persentase dari *width* tampilan jendelanya. Tag <HR> ini juga tidak memiliki tag penutup.

### **(6) Huruf *Bold*, *Italic*, *Under Line*, *Subscript*, *Superscript*, dan *Stripe***

Huruf dalam *web* menggunakan HTML dapat dibuat dengan beberapa efek cetak huruf dalam dokumen. Berikut ini untuk membuat beberapa efek cetak huruf dokumen:

- (a) Untuk memberikan efek cetak huruf tebal adalah (*bold*): <b>cetak tebal</b>.
- (b) Untuk memberikan efek cetak huruf miring adalah (*italic*): <i>cetak italic</i>.
- (c) Untuk memberikan efek cetak dengan huruf garis bawah adalah (*underline*): <u>underline</u>.
- (d) Untuk memberikan efek cetak huruf sebagai *subscript* : <sub>cetak subscript</sub>.
- (e) Untuk memberikan efek cetak huruf sebagai *superscript* : <sup>superscript</sup>.
- (f) Untuk memberikan efek huruf yang dicoret adalah (*strike*) : <s>cetak stripe</s>.

### **7) List**

Dokumen HTML dapat menampilkan daftar *list* seperti *Bullet* dan *Numbering* pada aplikasi program *Microsoft Word*. Ada dua macam *list* yaitu

*UnderList* (mirip *Bullets*) dan *OrderList* (mirip *Numbering*). Pada *UnderList* digunakan tag <UL> sebagai pembuka dan tag </UL> sebagai penutup, untuk *item-item list* yang ada didalamnya digunakan tag <LI> untuk mengawalinya. *Underlist* mempunyai atribut "type" yaitu bentuk dari *bullet*-nya. Ada beberapa jenis "type" yaitu *disc (default)* untuk lingkaran tidak berlubang, *circle* untuk bentuk lingkaran berlubang, *square* untuk kotak.

*UnderList* juga memiliki atribut tipe untuk menentukan tipe penomorannya, tipe-tipenya adalah:

- (a) 1 untuk penomoran 1,2,3 dan seterusnya.
- (b) A untuk penomoran A, B, C dan seterusnya.
- (c) I untuk penomoran I, II, III, dan seterusnya.
- (d) a untuk penomoran a, b, c, dan seterusnya.
- (e) i untuk penomoran i, ii, iii, dan seterusnya.

Bila tipe penomoran tidak didefinisikan tipenya, secara *default OrderList* akan muncul penomoran angka 1,2,3 dan seterusnya.

## **8) Fontasi**

Pemrograman untuk menghasilkan cetak huruf yang bermacam-macam karakter dengan memodifikasi *font* digunakan tag <FONT> sebagai pembuka dan tag </FONT> sebagai penutup. Tag ini mempunyai beberapa atribut, antara lain untuk atribut "size" untuk menentukan ukuran besar huruf, atribut "color" untuk menentukan warna. Menentukan warna dapat menggunakan aturan RGB atau dengan *pre-define color*, sedangkan untuk jenis hurufnya dengan atribut "face".

Nilai atribut *face* dapat juga diberikan alternatif *type font* pengganti dengan dibatasi tanda koma. Misal : `<FONT color="#FF0000" size="7"face="verdana,arial">`. Dalam tag *font* ini dapat diberikan juga tag untuk mengatur efek cetak huruf tag `<B>`, `<I>` dan sebagainya.

## 9) *Image*

Dalam dokumen HTML dapat ditampilkan gambar (*image*) dengan menggunakan tag `<IMG SRC="nama file gambar">`. Tag ini mempunyai beberapa atribut antara lain:

- (a) *Width* digunakan menentukan lebar gambar.
- (b) *Height* digunakan menentukan tinggi gambar.
- (c) *Border* digunakan untuk menentukan bingkai pada gambar.
- (d) *Hspace* digunakan untuk memberi ruang kosong disebelah kiri dan kanan gambar.
- (e) *Vspace* digunakan untuk memberi ruang kosong di atas dan di bawah gambar.
- (f) *Align* digunakan untuk menentukan perataan gambar, di kiri, di tengah, di kanan, di atas, di bawah pada layar.
- (g) *Alt* digunakan untuk memberi komentar bila *pointer* berada diatas gambar.

Tag `<IMAGE>` ini tidak memiliki tag penutup. Jika gambar tidak berhasil ditampilkan perlu dicoba memindahkan letak *file* gambar yang hendak ditampilkan ke *folder* di mana *file* .htm atau .html berada.

## 10) *Hyperlink*

Untuk menghubungkan link antar web yang satu dengan web yang lain digunakan *hyperling* atau *ancor*, dalam satu *web site* atau antara *web site* atau juga untuk berpindah ke posisi dalam halaman *web site*. Tag ini mempunyai sifat *link*, *visited*, dan *active*. Deskripsi sifat tersebut adalah:

- (a) *Link*, saat *hyperlink* belum dikunjungi.
- (b) *Visited*, *hyperlink* telah dikunjungi.
- (c) *Active*, *hyperlink* dalam keadaan in fokus atau terpilih.

Sifat-sifat tag ini dapat diatur melalui atribut dalam *body* bukan atribut tag *hyperlink*, yaitu dengan cara sebagai berikut:

```
<BODY link="#...."vlink="#...."alink=#....">
```

Tag yang digunakan untuk membuat *hyperlink* adalah tag <A HREF="tujuan"> sebagai pembuka dan tag </A> sebagai penutup.

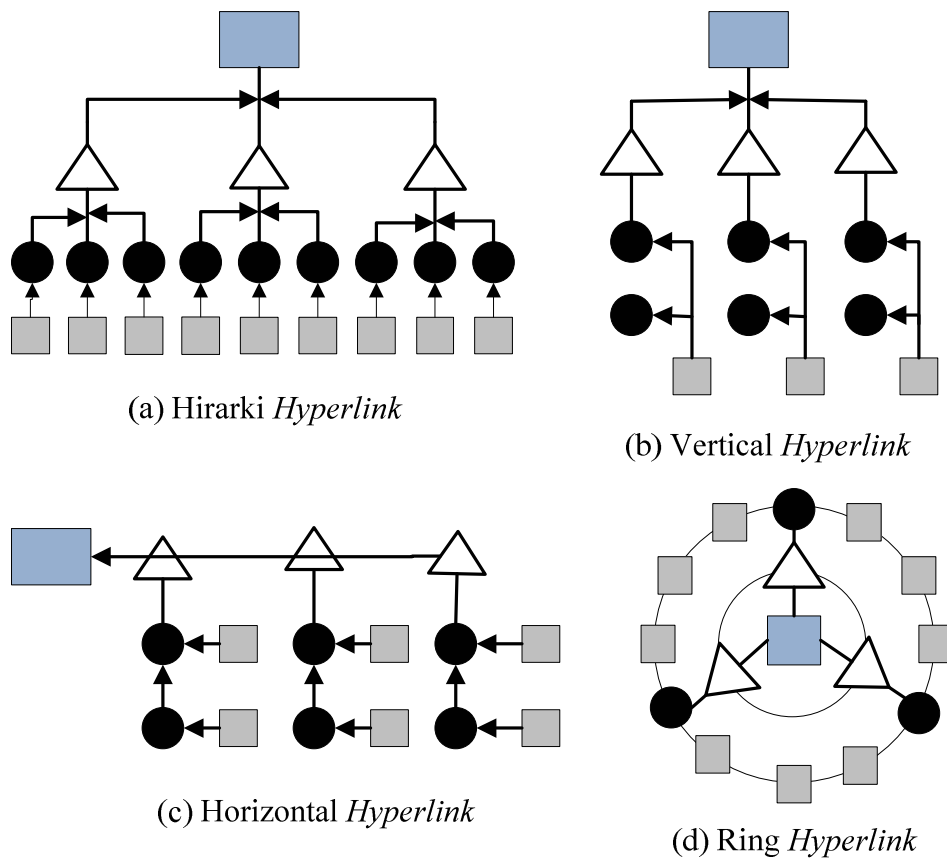
Tujuan *hyperlink* dapat diisikan dengan ketentuan sebagai berikut:

- (a) "name file" perpindahan antara halaman dalam satu *web site*.
- (b) "http://URL" perpindahan antara halaman dalam *web site* atau *server* yang berbeda.
- (c) "#posisi" perpindahan posisi dalam halaman yang sama (*bookmark*).
- (d) "mailto:[alamatemail]" berpindah ke *email compose* dengan alamat yang diberikan.

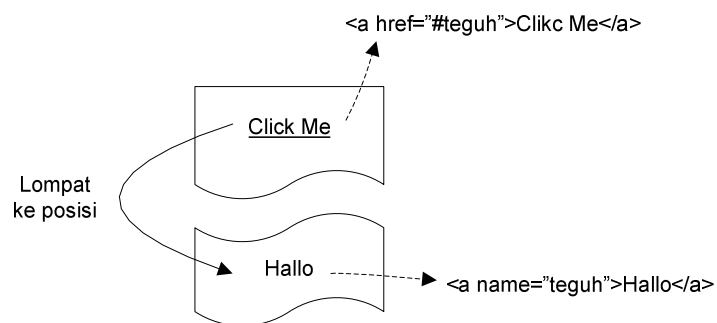
*Hyperlink* dapat berupa kata atau kalimat yang diketikkan antara tag `<A HREF="tujuan">.....</A>` pada tampilan *browser* nya akan digaris bawah kata atau kalimat tersebut.

Dapat juga *hyperlink* berupa gambar, susunan tag nya menjadi `<A HREF="tujuan"><IMG SRC="..."></A>`. Setiap *hyperlink* bila kursor didekatkan di atasnya akan berubah menjadi bentuk jari, jika di klik akan dihubungkan ( *di link*) ke alamat tujuan.

Perpindahan posisi dalam halaman yang sama terlebih dahulu dibuat penanda posisi dalam halaman tersebut dengan menggunakan tag `<A name="#...">` pada posisi halaman yang diinginkan tanpa diberi tag penutup. Misal *hyperlink* terhadap posisi `<A name="teguh"> Click Me</A>` dilakukan dengan memberikan tag `<A hyperlink="teguh">Click Me</A>`. Sehingga kalimat Click Me diklik posisi akan menuju ke penanda *teguh*, lihat gambar 2.4.



**Gambar 2.3** Beberapa bentuk struktur *hyperlink* (Suteja BR, 2007. h.19-20)



**Gambar 2.4** Pindah posisi dalam halaman (Suteja BR, 2007. h.20)

Perpindahan posisi dengan menggunakan atribut target pada *hyperlink* masih ada beberapa cara untuk berpindah dengan atribut target dapat menampilkan *link* halaman pada posisi tampilan tertentu.

Misal : `<a href="sipak.dosen.undip.info"target="_blank">Klik</a>`

*Web applications* "sipak.dosen.undip.info" akan muncul dalam jendela baru jika *hyperlink* dijalankan.

### **11) Atribut Tabel**

Dalam pengisian penetapan angka kredit jabatan dosen, tabel merupakan elemen penting dalam pengisian angka kredit, sebab jika tidak menggunakan tabel akan terjadi kesulitan dalam pengisian dokumen. Tag untuk membuat tabel dibagi menjadi dua berdasarkan pada letak tag. Tag *outside* yang digunakan adalah `<TABLE></TABLE>` sebagai tag dasar, sedangkan *inside* adalah tag `<TR></TR>` untuk menyatakan setiap baris yang ada pada tabel tersebut. Tag `<TR>` berada dalam tag `<TABLE>`. Dan tag `<TD></TD>` untuk menyatakan kolom pada setiap baris dalam tag `<CAPTION>....</CAPTION>` dapat digunakan untuk menampilkan judul tabel.

Sebaiknya ukuran tabel dapat diperbesar dan dapat diperkecil (*resizeable*), bagian tabel dapat di isi informasi (tulisan, gambar, atau obyek lain termasuk membuat tabel lagi) yaitu berada dalam kolom atau didalam tag `<TD>` ,karena diperbolehkan membuat tabel dalam tabel (*nested table*).

### **12) Katakunci (Password)**

Pengetikan kata kunci atau *password* setiap karakter akan disembunyikan dengan melakukan *masking* karakter ke \* atau •. Tetapi paling sering digunakan untuk input *account password* atau *pin code*, seperti:

```
<INPUT type="PASSWORD" name="Pwd" size="8" maxlength="8".
```

### 13) *Hidden*

*Hidden* adalah obyek yang tidak dapat dilihat tampilannya secara langsung dalam web *browser*. Obyek tipe ini dapat digunakan bertukar nilai sementara antar variabel dalam *server programming*. Sifat obyek ini adalah statis, artinya tidak dapat diganti secara langsung. Contoh sintaksnya:

```
<INPUT type="HIDDEN" name="kunci" value="rahasia">
```

### 14) *Select*

Mode *select* ini digunakan untuk memilih suatu daftar yang disajikan dalam menu (*combo box*, *spinner*, dan *menu list*). Tag yang dipakai untuk pendefinisian daftar pilihan adalah tag <SELECT>. Untuk daftar pilihan atau opsi didefinisikan dengan tag <OPTION> dan berada dalam tag <SELECT>.

### 15) *Frame*

Halaman *web applications* tampilan areanya terbagi-bagi. Konsep *frame* mirip dengan tabel, perbedaannya masing-masing bagian kolom atau baris dalam *frame* adalah merupakan *file* sedangkan dalam tabel adalah sel. Dapat dikatakan

suatu *frame* merupakan kumpulan banyak *file* dokumen HTML yang tersusun sedemikian rupa.

Membuat frame ditentukan dari atribut yang digunakan:

- (a) Lebih dari satu area dapat ditampilkan secara bersamaan dalam satu tampilan dokumen.
- (b) Setiap dokumen ditampilkan dalam sebuah *frame*, minimal satu *frame* memiliki satu file.
- (c) Ada dokumen tersendiri yang berisi definisi *frameset*. *Frameset* membungkus keseluruhan *file frame*. Dokumen *frameset* tidak mempunyai isi dokumen (tidak ada <body></body>)
- (d) Skema dasar *frameset*:

```
<FRAMESET          FRAMEBORDER="x" BORDER="y" FRAME
SPACING="z">
  <FRAME src="namafile1.html" name=".....">
  <FRAME src="namefile2.html" name=".....">
  .....dan seterusnya (atau frameset yang lain)
<NOFRAMES>
Bagian ini ditampilkan jika browser tidak mendukung frame
</NOFRAME>
</FRAMESET>
```

- (e) *Frameset* menentukan tata letak, ukuran dan banyaknya *frame* yang akan ditampilkan. Untuk menentukan tata letak dokumen dalam *frame* yang tersusun secara baris ditambahkan atribut:

rows="ukuran-1, ukuran-2,.....ukuran-n.

Sedangkan untuk kolom digunakan atribut:

cols="ukuran-1, ukuran-2,.....ukuran-n.

Pendefinisian ukuran frameset digunakan aturan sebagai berikut:

- (a) Nilai angka dalam *pixel*.
- (b) Nilai angka dalam persentase (%) yang menandakan persentase dari keseluruhan area tampilan yang disediakan. Contoh:

```
<FRAMESET rows="50,10%,*,20"> (tampilan terbagi 4 baris)
```

```
.....  
</FRAMESET>
```

```
<FRAMESET cols="30%,*"> (tampilan terbagi 2 kolom)
```

```
.....  
</FRAMESET>
```

Tanda bintang (\*) menandakan sisa ruang yang masih ada

- (f) Dengan menentukan ukuran-ukuran tersebut, *frameset* secara otomatis terbagi menurut area tampilan sebanyak yang telah ditentukan.
- (g) *Frame name* digunakan sebagai *hyperlink* untuk mengaktifkan *link* ke *frame* tertentu yang sesuai dengan target *name*-nya.
- (h) Di dalam *frameset* boleh terdapat *frameset* lain.

*Frame* yang terlalu banyak akan membuat pengguna menjadi sibuk menggeser (*scrolling*) halaman *web*, sehingga perlu memperhitungkan kenyamanan pengguna dalam mengaplikasi di area halaman *web*.

## 16) Button (Tombol)

Tombol atau biasa disebut *Button* digunakan untuk melakukan perintah eksekusi dari semua komponen yang telah dibuat sebelumnya. Tombol dapat dibedakan menjadi 3 macam: *Submit*, *Reset*, dan *None(button)*.

### **2.2.3 Desain Aplikasi Web dengan PHP**

#### **1) Memulai program PHP**

Salah satu kelebihan PHP dapat disisipkan kedalam tag-tag HTML, istilah dari pengertian ini adalah bahasa yang dapat *embedded* di dalam bahasa program lain. Tetapi PHP juga dapat berjalan sendiri tanpa berada disela-sela program yang lain.

Selain *embeded*, PHP juga dapat dibuat menggunakan format tag PHP murni, yaitu dengan mengubah semua tag HTML menjadi script PHP dengan menghilangkan tanda petik (“) yang ada pada setiap atribut tag HTML.

#### **2) Menampilkan data dalam program PHP**

Dalam program PHP, kelompok informasi yang hendak disajikan ke dalam halaman *browser* ditentukan oleh suatu perintah penampil, antara lain *echo* dan *print* seperti berikut:

```
<?php echo"informasi akan ditampilkan"?>
```

Atau

```
<?php print ("informasi yang ditampilkan"?>
```

#### **3) Variabel dalam PHP**

Pembentukan *variable* dalam PHP dengan memberikan tanda *String* (\$) sebagai pendeklarasian awal, dengan menggunakan tanda \$ karakter yang ada disebelahnya akan dianggap atau dikenali oleh program sebagai bentuk variabel.

Aturan-aturan yang dapat digunakan untuk menulis variabel adalah sebagai berikut:

- (a) Pendeklarasian variabel harus ditandai dengan *String* (\$).
- (b) Variabel dapat menggunakan harus kecil maupun harus besar atau perpaduan antara keduanya.
- (c) Penulisan awal variabel tidak boleh menggunakan angka.
- (d) Apabila ingin menggunakan angka sebagai variabelnya, untuk memulainya harus menggunakan tanda *underscore*(\_).
- (e) Tidak boleh menggunakan bentuk-bentuk karakter yang tergolong kode ASCII.

Di dalam praktik pemrograman PHP ada beberapa teknik penggunaan, antara lain: variabel biasa, variabel dalam sebuah kelompok, dan variabel antar halaman.

Variabel biasa bentuk penulisan dengan menggunakan tanda *String* yang diikuti oleh isi variabel atau nama variabelnya. Dari variabel-variabel tersebut dapat dibaca berulang kali dalam suatu halaman *web*.

Variabel dalam sebuah kelompok digunakan untuk mendeklarasikan data apabila menggunakan fungsi dalam PHP, isi yang ada dalam variabel tidak dapat ditampilkan apabila tidak melakukan pemanggilan fungsi yang mendeklarasikan variabel tersebut.

Variabel antar halaman digunakan untuk menghantarkan data yang ada dari setiap variabel ke dalam halaman lain, bentuk-bentuk variabel ini biasanya berasal dari data semacam form data seperti formulir pendaftaran atau yang berhubungan dengan formulir lainnya.

#### **4) Operator dalam PHP**

Pemrograman *web application* menggunakan PHP akan selalu menggunakan beberapa operator, karena pemrograman PHP telah menjadi bentuk pemrograman yang kompleks, dengan pemrograman PHP ini dapat dihasilkan sebuah sistem yang mampu menangani banyak permasalahan dalam *web*. Beberapa operator dalam PHP telah disetandarkan dengan bentuk pemrograman yang lain, diantaranya adalah: operator aritmatik, operator penaikan dan penurunan, operator logika, dan operator pembandingan.

#### **5) Pengkondisian**

Pengkondisian diartikan sebagai suatu bentuk logika yang menganalogikan dengan sesuatu kejadian atau beberapa kejadian. Pada setiap kejadian yang ada, kondisi akan dibaca yang selanjutnya akan diteruskan dengan bentuk ungkapan yang menyatakan hasil dari kondisi tersebut.

Dalam pemrograman PHP, pengkondisian ini merupakan bentuk dasar untuk membangun logika yang harus diketahui karena pemrograman memiliki logika dan algoritma. Apabila telah memahami logika dan algoritma akan

memudahkan dalam menerapkan pada suatu program. Ada beberapa bentuk kondisi dalam pemrograman PHP yang perlu diketahui:

- (a) Kondisi satu pilihan, kondisi ini hanya diberikan satu pilihan atau satu kondisi, dan apabila kondisi memenuhi akan menghasilkan proses berikutnya. Untuk mengkondisikan ini digunakan operator *if*, misal sebagai berikut:

**if (kondisi)  
Pernyataan**

Pada sintaks diatas bila kondisi terpenuhi, proses akan menuju kepernyataan, bila kondisi tidak terpenuhi maka proses tidak akan dikerjakan atau berhenti

- (b) Kondisi dua pilihan, disini menggunakan dua pilihan yang berbeda, dengan ketentuan bila ketentuan yang pertama dianggap salah atau tidak sesuai, program akan melanjutkan pada kondisi berikutnya, hal tersebut digunakan *if-else*.
- (c) Kondisi tiga pilihan atau lebih, menentukan tiga pilihan atau lebih yang dijadikan kondisi, hal ini digunakan bila terjadi banyak permasalahan, dengan ketentuan tiap permasalahan mempunyai nilai yang berbeda. Untuk menyelesaikan permasalahan tersebut menggunakan *if-else if-else*.
- (d) Kondisi SWITCH, sebagai pengganti bentuk *if-then-else*, kedua fungsi ini mempunyai bentuk algoritma yang dapat dikatakan hampir sama.

Hanya saja dengan SWITCH dapat membuat sebuah pengkondisian dengan hasil ekspresi lebih mudah dan cepat.

## 6) *Looping*

*Looping* adalah sebuah bentuk pengulangan sebuah pernyataan atau kejadian, fungsi ini sangat membantu dalam membuat program yang berkaitan pengulangan data tertentu. Ada beberapa fungsi di dalam *looping*, antara lain menggunakan atribut FOR, menggunakan atribut WHILE, menggunakan atribut do...WHILE, dan menggunakan atribut FUNGSI.

Atribut *for* merupakan bentuk perulangan sederhana, dengan menggunakan fungsi ini dapat melakukan pengulangan data sampai melampaui batas yang ditentukan, contoh sintaksnya sebagai berikut:

```
for (ekspresi1; ekspresi2; elspresi3)  
{  
pernyataan  
}
```

- Ekspresi1: merupakan sebuah nilai awal yang harus ada pada saat menggunakan *for*.
- Ekspresi2: digunakan sebagai batasan data yang akan diulang.
- Ekspresi3: merupakan bentuk pengulangannya, apakah data akan dilakukan penambahan, pengurangan, dan sebagainya.

Atribut *while* juga merupakan bentuk fungsi untuk pengulangan, pernyataan akan terus dikerjakan bila masih memenuhi pada ekspresi, dan bila kondisi tidak memenuhi pernyataan akan selesai, contoh sintaksnya adalah:

```
While (ekspresi)  
{  
pernyataan  
}
```

Atribut *do...while* hampir sama dengan *while*, yang membedakan dengan fungsi *do...while* adalah pernyataan akan dikerjakan terlebih dahulu sebelum melakukan pengecekan pada ekspresi, dan apabila memenuhi maka proses akan terus berjalan, dan bila tidak maka proses dihentikan, contoh sintaksnya sebagai berikut:

```
do {  
pernyataan  
}  
while (ekspresi)
```

Atribut FUNGSI adalah metode PHP untuk mengelompokkan program sehingga pada program tersebut akan dipecah menjadi bentuk-bentuk program kecil yang dikelompokkan dengan menggunakan bentuk program utama. Untuk mengelompokkan program tersebut hanya memanggil FUNGSI nya saja, sehingga pada program tersebut telah memiliki data seperti yang dideklarasikan pada fungsi tersebut.

Pada saat pembentukan sebuah FUNGSI, semua variabel data yang akan digunakan dalam FUNGSI harus didefinisikan semua dengan menggunakan *type global*, contoh sintaksnya sebagai berikut:

```
function nama_fungsi ()  
{  
global $variable,.....,$variable_n;  
pernyataan  
}
```

Dari FUNGSI diatas bila ingin memanggilnya hanya menuliskan nama atribut FUNGSI tersebut ke dalam program utama.

#### 2.2.4 Desain Aplikasi Web dengan JavaScript

Penulisan *JavaScript* diletakkan antara tag HTML, dengan menambahkan *JavaScript* akan membuat halaman *web* menjadi lebih interaktif. Menggunakan *JavaScript* memungkinkan kostominasi terhadap HTML pada saat diakses dengan menulis melalui penanganan *event* terhadap elemen-elemen tag HTML dalam halaman tersebut, memeriksa data *form* pada sisi klien hingga membuat animasi kursor *mouse* dan lain sebagainya.

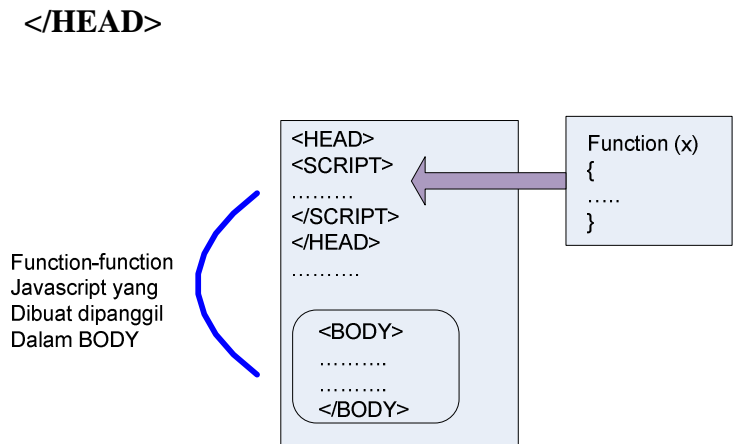
*JavaScript* bersifat *Case Sensitive*, yaitu membedakan huruf besar dengan huruf kecil, untuk setiap perintah dan penamaan variabelnya, seperti contoh variabel **Coba** berbeda dengan varibel **coba**. Contoh sintaks penulisan *JavaScript* dalam HTML dapat dilihat pada gambar 2.5.

(a) *Body* dalam dokumen HTML

```
<BODY>
  <script language="javascript">
    //baris kode javascript diketik disini
    //setiap perintah diakhiri tanda ;
  </script>
</BODY>
```

(b) *Header* dalam bentuk *function*

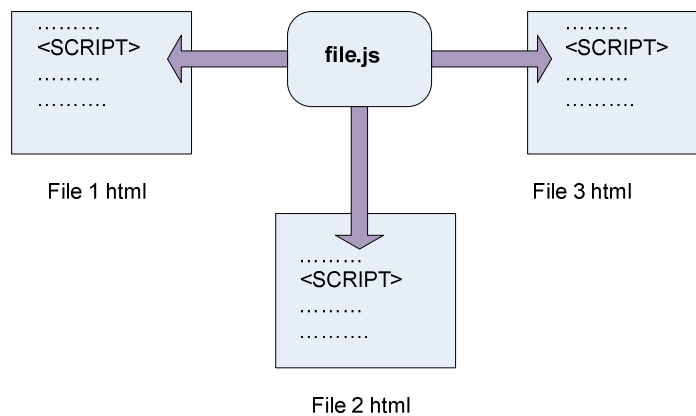
```
<HEAD>
<script language="javascript">
Function namafunction(parameterinput)
{
    //baris kode fungsi disini dan diakhiri oleh ';'
}
</script>
```



**Gambar 2.5** Internal JavaScript pada HTML (Suteja BR, 2007. h.102)

- (c) Penulisan terpisah dari HTML (*external Javascript*), lihat gambar 2.6. Bilamana JavaScript diletakkan pada file (file.js) terpisah maka file tersebut dapat dipanggil dengan penulisan:

```
<SCRIPT language="javascript"src=path/file.js"></SCRIPT>
```



**Gambar 2.6** External JavaScript pada HTML (Suteja BR, 2007. h.102)

### 2.2.5 Desain Web Applications dengan CSS

CSS mendukung sifat-sifat yang berbeda yang dapat diterapkan sebagai penyeleksi untuk menentukan presentasi dari sebuah dokumen HTML. Properties

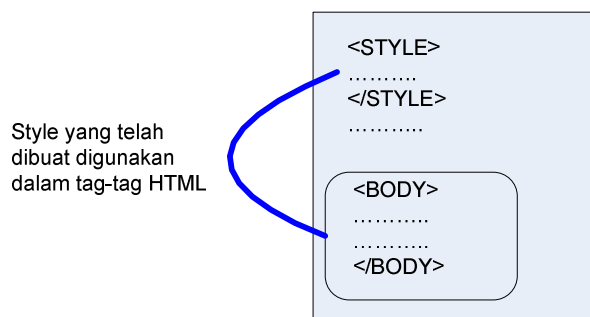
meliputi latar belakang (warna atau grafik), ukuran *font*, *font-weight*, *line-height* (jarak terkemuka atau *interlinear*), *font-family*, surat-*spacing* dan *word-spacing*.(Richmond, A.,tth)

CSS dapat digunakan untuk menampilkan *style* dari *content* dokumen HTML lebih menarik dengan meminimalkan tag-tag HTML, sehingga menghemat waktu pengerjaannya dan lebih cepat proses *loading*-nya oleh *web browser*. Secara singkat CSS akan mempercantik penampilan dokumen HTML.

Penulisan *style* (CSS) cukup didefinisikan untuk dapat diterapkan di semua *web applications* yang diinginkan. Sehingga proses pengaturan tampilan *web* (perubahan *style*) secara otomatis. Ada 3 cara dalam mengimplementasikan CSS, yaitu sebagai berikut:

- (a) *Inline style*, yaitu menjadi atribut dalam tag HTML, atributnya adalah *style* didefinisikan sebagai *property* dengan sintaks sebagai berikut  
**style="property-1 nilai; property-2: nilai;.....;property-n:nilai"**
- (b) *Internal style sheet* atau *embed style* yaitu berada didalam tag <HEAD> dan didefinisikan seperti gambar 2.7.

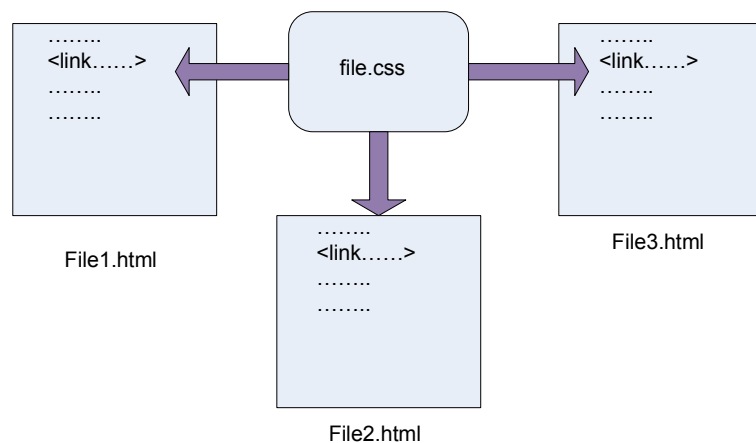
**<STYLE type=tex/css>.....</STYLE>**



**Gambar 2.7** *Internal style sheet* (Suteja BR, 2007. h.60)

(c) *External style sheet* yaitu CSS berda terpisah pada sebuah *file.css*, lihat gambar 2.8. Untuk memanggilnya dalam dokumen HTML menggunakan atribut link, pemanggilan berada dalam HEAD, contoh sintaksnya sebagai berikut:

```
<link rel="stylesheet" type="text/css" href="file.css"/>
```



**Gambar 2.8** *External style sheet* (Suteja BR, 2007. h.61)

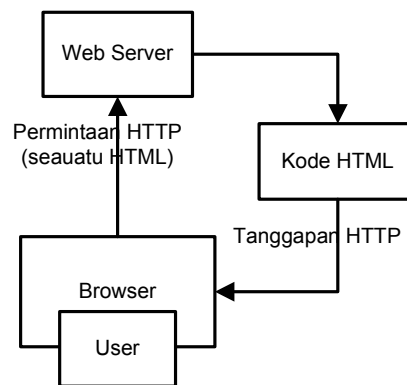
Sintaks *style* untuk *internal* atau *external* CSS adalah sebagai berikut:

```
selector{
    property-1:nilai;
    property-2:nilai;
    .....
    Property-n:nilai
}
```

*Selector* adalah tag HTML yang akan dikenal CSS, *property* adalah atribut yang ingin diterapkan pada tag HTML. Pada masing-masing *selector* yang dipakai dapat memiliki banyak *property* beserta nilai atau *value*-nya dengan dibatasi tanda ; (titik koma).

### 2.3 Pengertian Web Server

*Web server* hanya menyediakan layanan atau *service*. Layanan digunakan atau dipanggil oleh pengguna, *user*, dan yang menjadi *interface* adalah aplikasi yang memanggilnya dan bukan *web server*nya. Lihat gambar 2.9.



**Gambar 2.9** Skema akses web server (Kadir, 2003, h.5)

*Web server* berisikan *database*, misalnya data dosen, dalam pemanggilannya dengan layanan *web* dapat dipadukan untuk isian usulan angka kredit jabatan fungsional.

Pengguna, *user*, untuk menggunakan *web* harus mempunyai *password*, kata kunci, jika *password* cocok dan sesuai yang diisikan dalam *database login* pemakaian *web* akan muncul sesuai harapan, bilamana *password* tidak cocok akan ditolak.

### 2.4 Diagram Alir Perangkat Lunak

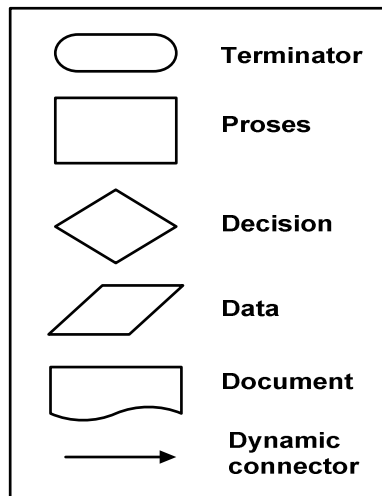
Menurut *edrawsoft.com*, untuk masuk ke *server* sebaiknya menuruti diagram alir, diagram alir program *web* dan diagram teknis. Simbol diagram alir atau *flowchart* akan membantu untuk membuat diagram alir yang paling umum.

Simbol *Flow Diagram Standar Flowchart* menggunakan bentuk khusus untuk mewakili berbagai jenis tindakan atau langkah dalam proses. Garis dan panah menunjukkan urutan langkah-langkah, dan hubungan di antara mereka.

*Flow diagram* adalah jenis *organizer grafis* yang menampilkan visual rantai instruksi yang digunakan untuk menyelesaikan suatu algoritma atau proses rumit lain.

Diagram alir memiliki awal, hasil yang mungkin beberapa di beberapa *node*, peraturan pada beberapa *node*, dan akhir beberapa kemungkinan. Dalam diagram alir, simbol yang berbeda memiliki arti yang berbeda. Simbol-simbol diagram alir dapat dilihat pada gambar 2.10.

- Lingkaran atau oval mulai, berhenti, atau kontrol poin.
- Empat persegi panjang sebagai proses.
- Berlian poin keputusan.
- *Rectangle* dan kotak adalah langkah di mana proses berlangsung.
- Jajaran genjang mewakili data input atau output.
- Panah mewakili arah aliran



**Gambar 2.10** Simbol-simbol diagram alir

Untuk membangun sebuah diagram alir yang efektif paling tidak harus memenuhi beberapa ketentuan sebagai berikut:

- (a) Tentukan batas-batas proses dengan awal dan titik akhir.
- (b) Lengkapi gambaran besar sebelum mengisi rincian.
- (c) Jelas mendefinisikan setiap langkah dalam proses. Akurat dan jujur.
- (d) Mengidentifikasi waktu tertinggal dan non-nilai tambah langkah.
- (e) Mengedarkan diagram aliran kepada orang lain yang terlibat dalam proses untuk mendapatkan pendapat mereka.

Diagram alir tidak dapat bekerja jika tidak akurat, dan harus benar dalam proses dan bebas untuk menggambarkan apa yang sebenarnya terjadi. Sebuah *flowchart* menyeluruh harus memberikan pandangan yang jelas tentang bagaimana proses pembuatan *web* dan proses user menggunakan aplikasi web.

## **2.5 Data Flow Diagram (DFD)**

*Data Flow Diagram* (DFD) atau disebut juga Diagram Alir Data (DAD) sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan tanpa mempertimbangkan lingkungan fisik dimana data mengalir atau lingkungan fisik dimana data tersebut akan disimpan. Dalam DFD dibahas fungsi- fungsi apa saja yang diperlukan oleh suatu sistem dan aliran data yang terdapat diantara proses didalamnya. DFD dapat dikembangkan dari level yang paling rendah ke level yang paling tinggi.

Kamus Lengkap Dunia Komputer (2002: 81) menyatakan DFD adalah suatu alat dalam perancangan yang menggunakan simbol – simbol untuk menggambarkan aliran data melalui serangkaian proses yang saling berhubungan.

**Tabel 2.1** Simbol-simbol *Data Flow Diagram* (DFD)

No	Nama	Simbol	
		Gano&Sarson	De Marco/Yourdan
1	Agen Eksternal		
2	Proses		
3	Aliran Data		
4	Data Store		

Ada beberapa tahapan dalam pembuatan DFD yaitu :

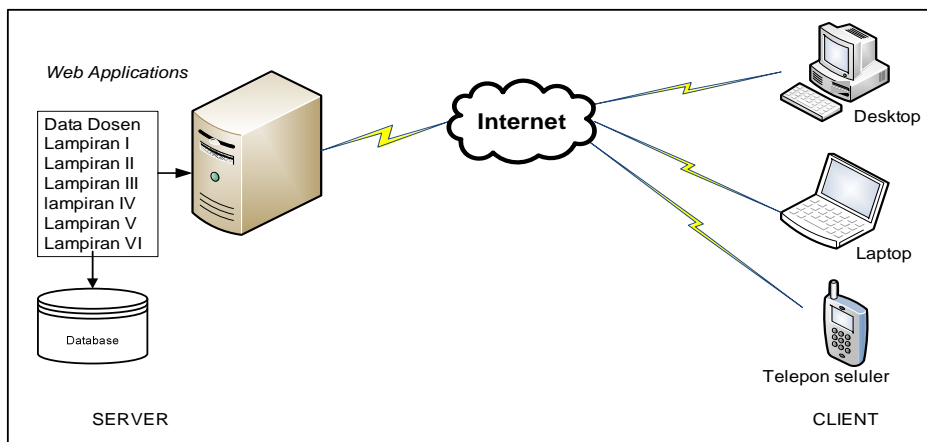
- (a) Diagram konteks; digunakan untuk menggambarkan sistem secara global
- (b) Diagram Level Nol; digunakan menggambarkan tahapan-tahapan proses yang ada dalam diagram konteks.

(c) Diagram Detil (level satu); digunakan untuk menggambarkan arus data yang lebih mendetail dalam proses diagram level nol.

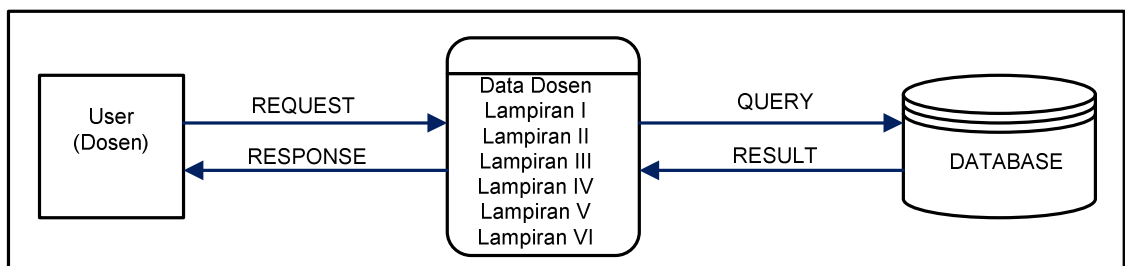
Symbol-simbol yang digunakan dalam (*Data Flow Diagram*) DFD seperti terlihat dalam tabel 2.1

## 2.6 Akses Aplikasi Web Usulan Angka Kredit

Akses sistem informasi penetapan angka kredit Dosen (SIPAK Dosen) lewat Internet ke server aplikasi web. Akses tersebut dapat dilakukan menggunakan komputer desktop, laptop, ataupun telepon seluler asalkan ada jaringan Internet seperti terlihat pada gambar 2.11.



**Gambar 2.11** Aplikasi web dapat diakses berbagai jenis aplikasi (Lucky, 2008, h.3)



**Gambar 2.12** Proses *request* dan *response* ke aplikasi web (Lucky. 2008. h.10)

Pengguna, *user*, me-request ke *database* terus meminta *query* ke aplikasi web di *server* dan akan me-result sesuai tampilan yang diminta oleh *user*, *user* akan melakukan pengisian di aplikasi web sesuai keinginannya, saat pengisian terjadi proses dalam database dan *response* dapat diamati langsung oleh *user*. Jika tampilan telah sesuai dengan keinginan *user*, keputusan selanjutnya ada pada *user*.

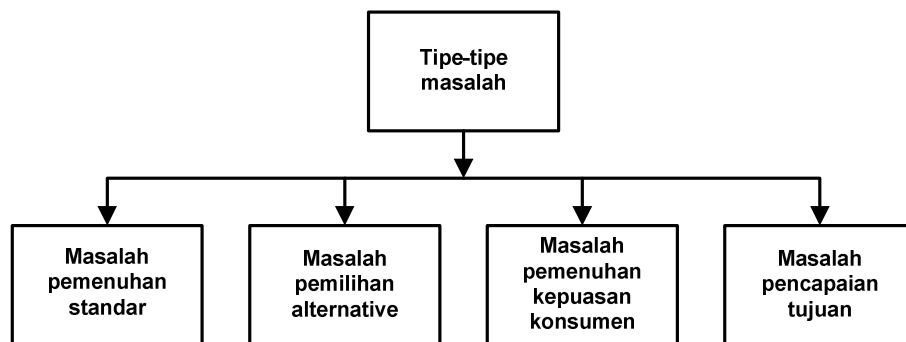
## BAB 3

### METODE PENELITIAN

#### 3.1. Cara Penelitian

Penelitian diawali dengan studi pustaka yang berhubungan dengan pemrograman *web* dan kenaikan jabatan fungsional dosen, menentukan bahan penelitian, membangun desain *web* dan mengaplikasikan dengan usulan kenaikan jabatan fungsional dosen.

Untuk membangun suatu *web applications* tidak lepas dari suatu masalah, masalah mempunyai beberapa tipe antara lain seperti terlihat pada gambar 3.1.



**Gambar 3.1** Tipe-tipe masalah (Mulyanto, 2008. h.21)

Masalah pemenuhan standar merupakan masalah-masalah yang berhubungan dengan pencapaian standar yang telah ditentukan sebuah organisasi W3C. dan berlaku dalam jangka relatif panjang.

Masalah pemilihan alternatif berhubungan dengan bagaimana memilih solusi terbaik dari berbagai alternatif berdasarkan kriteria-kriteria tertentu. Permasalahan ini dijumpai dalam memilih bahasa pemrograman yang cocok untuk membangun *web applications*.

Masalah pemenuhan kepuasan konsumen sering kali muncul, bahkan tidak hanya konsumen tetapi juga yang melakukan pemrograman itu sendiri sering terjadi ketidakpuasan. Konsumen memiliki berbagai macam keinginan yang berbeda. Untuk memenuhi seluruh keinginan konsumen yang berbeda sangat tidak mungkin dan sangat memberatkan. Oleh karena itu dicari suatu pemecahan yang baik untuk memenuhi standar kebutuhan konsumen.

Masalah pencapaian tujuan sering kali berubah-ubah, sering tidak tetap, sehingga masalah pencapaian tujuan bersifat jangka pendek (relatif).

### 3.2. Bahan Penelitian

Bahan-bahan penelitian diambil studi pustaka meliputi buku teks yang ada hubungannya dengan pemrograman *web*, peraturan-peraturan mengenai jabatan fungsional dosen, jurnal ilmiah yang ada hubungannya dengan pemrograman *web*, situs di internet yang ada hubungannya dengan pemrograman *web* dan kenaikan jabatan fungsional dosen, data lain yang relevan dengan judul penelitian.

### 3.3. Alat Penelitian

Pemakaian alat digunakan *computer sever* tempat database yang dapat diakses oleh user lewat internet memakai *personal computer*, *laptop* dan telepon seluler. Pemakaian alat antara lain dapat dilihat pada tabel 3.1.

**Tabel 3.1** *Kebutuhan perangkat keras*

No.	Nama Alat	Spesifikasi	Keterangan
	Komputer Server		Dapat menggunakan fasilitas
1	CPU	Intel Atom 2.2GHz	

2	<i>Harddisk</i>	320 GB	komputer yang telah beroperasi
3	RAM	2 GB	
4	Rpm	7200	
Alat User			Mempunyai fasilitas untuk internet
1	<i>Personal Computer</i>	-	
2	<i>Laptop</i>	-	
3	Telepon seluler	-	

Perangkat lunak yang digunakan adalah:

- Sistem operasi Windows XP
- Apache 2.2.14 (Win32) Open SSL 0.9.81
- Basis data MySQL versi server 5.0.45-community
- Bahasa pemrograman web PHP versi 5.3.1
- Server at localhost Port 80
- Windows Internet Explorer 8 versi 8.0.7600.16385 chipper 256 bit

### 3.4. Jalan Penelitian

Alternatif pemecahan masalah yang berhubungan dengan dosen agar dapat mudah untuk “*mencicil*” pengisian angka kredit jabatan fungsional menggunakan komputer dengan fasilitas isian penetapan angka kredit melalui Internet, dan yang paling mendukung adalah aplikasi web, untuk itu harus dibangun terlebih dahulu aplikasi web sebagai fasilitasnya.

Menurut *prbd.net*, ada 4 Langkah Proses Pengembangan *Web Applications* (aplikasi web). Empat langkah proses untuk pengembangan aplikasi *web* dianggap sebagai yang terbaik dan diterima di seluruh dunia oleh perusahaan pengembangan *web* untuk memberikan pelayanan yang lebih baik kepada pengguna.

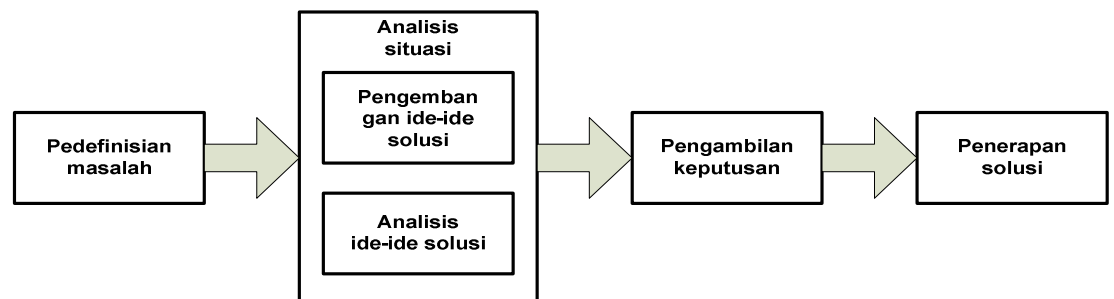
- a) Langkah pertama sebuah perusahaan pengembangan web untuk pengembangan aplikasi visualisasi dan aliran proyek, tim pengembangan proyek bekerja bersama menetapkan tujuan proyek termasuk penjadwalan, identifikasi keterbatasan dan aplikasi *upgrade*.
- b) Langkah kedua perencanaan yang berkaitan dengan berbagai aspek pengembangan aplikasi web. Rencana fitur yang harus dimasukkan dalam aplikasi dan *platform* apa yang harus digunakan untuk membangun *web*. Fase ini juga melibatkan pekerjaan pengalokasian pengembang untuk pengembangan aplikasi *web* secara keseluruhan.
- c) Langkah ketiga adalah langkah pengembangan, pengembang bekerja untuk menentukan variabel data, *coding* prosedur dan entitas yang akan digunakan dalam proses. Dokumen yang dipersiapkan menunjukkan bagaimana semua ini akan dilakukan dan dikirim ke manajemen untuk diperiksa. Pengembang tidak bisa melanjutkan sampai mereka mendapat persetujuan dari manajemen pada dokumen yang disajikan.
- d) Langkah ke empat adalah pengujian, ini salah satu langkah yang paling penting yang harus diikuti untuk pengembangan aplikasi web. Tanpa pengujian tidak ada cara lain untuk menentukan apakah aplikasi yang dikembangkan dapat berfungsi atau tidak sesuai harapan klien. Pengembang *web* mengadopsi berbagai metode yang berbeda, alat dan teknik untuk melakukan tes dan kesalahan dalam aplikasi.

Keempat langkah proses tersebut minimal harus diikuti oleh pengembang *web*, suatu formula akan sukses menghasilkan tujuan hasil yang pasti untuk membantu layanan aplikasi *web* yang lebih baik.

### 3.5 Analisis Masalah dan Pemecahan Masalah

Sering terjadi kesulitan membedakan masalah dan gejala. Gejala merupakan suatu tanda atau petunjuk terjadinya suatu masalah, kalau sudah mengetahui masalah dan gejalanya dapat ditentukan solusinya.

Pemecahan masalah merupakan proses dimana suatu situasi diamati kemudian bila ditemukan adalah masalah dibuat penyelesaiannya dengan cara menentukan masalah, mengurangi atau menghilangkan masalah atau mencegah masalah tersebut terjadi, lihat gambar 3.2.



**Gambar 3.2** Pemecahan masalah (Mulyanto, 2008. H h.22)

Pedefinisian masalah untuk memahami awal dari seluruh proses pemecahan masalah, memahami masalah dengan baik dan menghilangkan bagian-bagian yang kurang penting.

Analisis situasi dengan mengembangkan ide-ide solusi dan menganalisis ide-ide solusi untuk membuat rencana pemecahan masalah dengan mencari berbagai cara penyelesaian yang mungkin diterapkan untuk membuat rencana pemecahan masalah. Pemecahan suatu masalah pada umumnya tidak hanya satu tetapi beberapa macam. Dari sekian banyak alternatif pemecahan masalah hanya dipilih salah satu yang berdasarkan persyaratan tertentu merupakan cara yang paling baik untuk menyelesaikan permasalahan. Setelah memilih cara pemecahan masalah dapat dibuat rencana kasar penyelesaian masalah dengan membagi masalah ke bagian-bagian yang lebih kecil, sehingga tahapan-tahapan utama penyelesaian masalah dapat dilaksanakan.

Pengambilan keputusan dengan cara merinci tahapan-tahapan yang telah terbagi menjadi bagian-bagian lebih kecil disusun dengan pembagian dan urutan secara rinci yang harus ditempuh dalam penyelesaian masalah.

Penerapan solusi dengan memeriksa dan menyampaikan hasil dari pemecahan masalah bertujuan untuk memeriksa apakah akurasi (ketepatan) hasil dari cara yang dipilih telah memenuhi tujuan yang diinginkan. Selain itu juga untuk melihat bagaimana daya guna cara yang dipilih.

Agar setiap komputer ada fasilitas isian penetapan angka kredit perlu dibangun form isian penetapan angka kredit melalui Internet yaitu *web applications*.

## **3.6 Membangun Aplikasi Web**

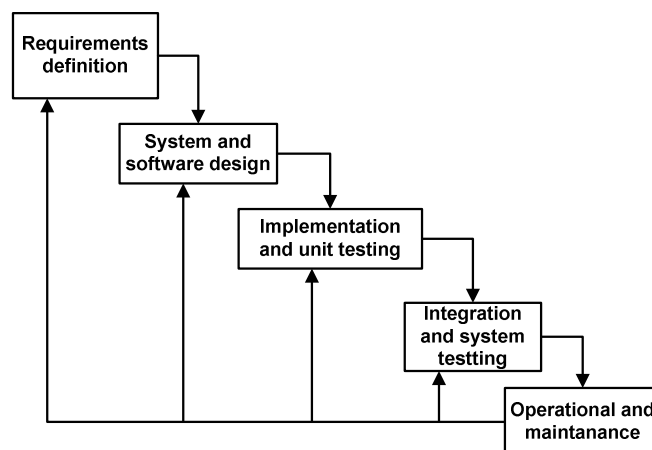
### **3.6.1 Langkah-langkah Membangun Aplikasi Web**

- a) Start merupakan awal yang akan diperoleh sehingga dapat menyiapkan segala sesuatu yang berhubungan dengan apa yang hendak dicapai.
- b) Identifikasi masalah apa yang terjadi, dari banyak masalah yang ada dapat diambil suatu masalah, disini diambil masalah dosen, yang berhubungan dengan penetapan angka kredit jabatan fungsional sebagai usulan kenaikan jabatan fungsional.
- c) Identifikasi keterbatasan hanya digunakan untuk dosen akademik, bukan dosen professional (vokasi), dan yang telah terdaftar di *server*.
- d) Variabel data merupakan semua angka kredit yang berhubungan dengan kinerja dosen yang akan dimasukkan dalam usulan penetapan angka kredit. Baik kegiatan pendidikan dan pengajaran, kegiatan ilmiah, kegiatan pengabdian pada masyarakat, dan kegiatan penunjang Tri Dharma Perguruan Tinggi.
- e) Rencana fitur aplikasi merupakan antarmuka *web* dengan pengguna untuk mempermudah memecahkan masalah dosen dalam pengisian angka kredit untuk usulan penetapan angka kredit jabatan fungsional sebagai syarat kenaikan jabatan fungsional.
- f) *Coding* merupakan bahasa pemrograman menggunakan HTML, PHP, *JavaScript*, dan CSS yang akan dipakai untuk membangun tampilan *web applications* agar dapat diisi oleh user dalam memasukkan data angka kredit kedalam komputer melalui jaringan Internet atau Intranet.

- g) Pengujian dilakukan setiap tahap selesainya *coding*, jika tampilan kurang memadai maka dilakukan peng-*coding*-an kembali agar mencapai hasil tampilan web yang lebih baik.
- h) Jika hasil pengujian *web* belum memadai sesuai tujuan perlu diperbaiki, kembali ke *coding* lagi dan di *upgrade*, diuji lagi agar sesuai tujuan yang diharapkan.
- i) Hasil akan diperoleh sesuai harapan tujuan setelah dilakukan beberapa kali pengujian dan dicoba dengan *entry* data dan tidak terjadi gejala yang menimbulkan masalah baru.

### 3.6.2 Metode Desain Aplikasi Web

Desain aplikasi *web* perlu dijabarkan secara lebih rinci agar dapat berwujud menjadi aplikasi *web* yang dapat berfungsi untuk pengisian angka kredit dosen. Menurut referensi (Sommerville.2001), rekayasa perangkat lunak dapat dilaksanakan dengan salah satu metode langkah-langkah mengalir seperti model air terjun (*waterfall model*) seperti terlihat pada gambar 3.3.



**Gambar 3.3** Rekayasa perangkat lunak model *waterfall* (Sommerville.2001)

Dalam *Waterfall*, seluruh proses pengembangan perangkat lunak dibagi menjadi fase-fase proses yang terpisah. Fase-fase dalam model *waterfall* adalah: *Requirements definition*, *System and software desig*, *Implementation and unit testing*, *Integration and System Testing*, dan *Operational and maintenance*. Semua fasa ini mengalir untuk satu sama lain sehingga fase kedua dimulai dan ketika didefinisikan seperangkat tujuan yang dicapai untuk tahap pertama dan itu jadi sempurna baru menginjak fase berikutnya dan harus sempurna juga, terus ke fase berikutnya lagi, demikian hingga fase terakhir. Setiap fase dapat dikontrol balik ke awal fase sebelumnya sehingga semua sistem fase sempurna semua, metode dan proses yang dilakukan dalam model *waterfall* menjadi suatu sistem lebih dapat terlihat sempurna semua.

Tahap *Model Waterfall* adalah:

- a) *Requirements definition*: Semua persyaratan yang mungkin dari sistem yang akan dikembangkan ditangkap dalam fase ini. Persyaratan yang ditetapkan fungsi dan kendala yang pengguna akhir (yang akan menggunakan sistem) mengharapkan dari sistem. Persyaratan yang dikumpulkan dari pengguna akhir melalui konsultasi, persyaratan ini dianalisis untuk validitas dan kemungkinan menggabungkan persyaratan dalam sistem yang akan pembangunan juga dipelajari. Akhirnya, dokumen spesifikasi kebutuhan dibuat yang melayani tujuan dari pedoman untuk tahap berikutnya dari model.
- b) *System and software design*: Sebelum memulai untuk *coding* yang sebenarnya, sangat penting untuk memahami apa kita akan membuat dan apa yang

seharusnya terlihat. Spesifikasi persyaratan dari tahap pertama yang dipelajari dalam fase ini dan desain sistem siap. Desain sistem membantu dalam menentukan persyaratan perangkat keras dan sistem dan juga membantu dalam menentukan arsitektur sistem secara keseluruhan. Spesifikasi desain sistem melayani sebagai masukan untuk tahap berikutnya dari model.

- c) *Implementation and unit testing*: Menerima dokumen desain sistem, pekerjaan ini dibagi dalam modul/unit dan pengkodean yang sebenarnya dimulai. Sistem ini pertama kali dikembangkan dalam program-program kecil yang disebut unit, yang terintegrasi dalam tahap berikutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas, ini disebut sebagai Unit Testing. Unit pengujian terutama memverifikasi jika modul/unit memenuhi spesifikasi mereka.
- d) *Integration and System Testing*: Seperti disebutkan di atas, sistem ini pertama dibagi dalam unit yang dikembangkan dan diuji untuk fungsi mereka. Unit ini diintegrasikan ke dalam sistem yang lengkap selama fase Integrasi dan diuji untuk memeriksa apakah semua modul/unit koordinasi antara satu sama lain dan sistem secara keseluruhan berperilaku sesuai spesifikasi. Setelah berhasil pengujian perangkat lunak, itu disampaikan kepada pelanggan.
- e) *Operational and maintenance*: fase model *waterfall* hampir tidak pernah berakhir (sangat panjang). Umumnya, masalah dengan sistem yang dikembangkan (yang tidak ditemukan selama siklus hidup pengembangan) muncul setelah penggunaan praktisnya dimulai, sehingga isu-isu terkait ke sistem yang dipecahkan setelah penyebaran sistem. Tidak semua masalah

datang dalam gambar secara langsung tetapi mereka muncul waktu ke waktu dan perlu diselesaikan, maka proses ini disebut sebagai pemeliharaan.

Desainer perangkat lunak mengumpulkan berbagai pandangan dan saran dari penguji yang dipilih dan menganalisis kembali untuk di *design* dan memodifikasi kode jika diperlukan. Setelah produk berhasil dalam testing, berarti program perangkat lunak sukses dalam pengujian penerimaan, produk dapat di *upload* untuk pelanggan.

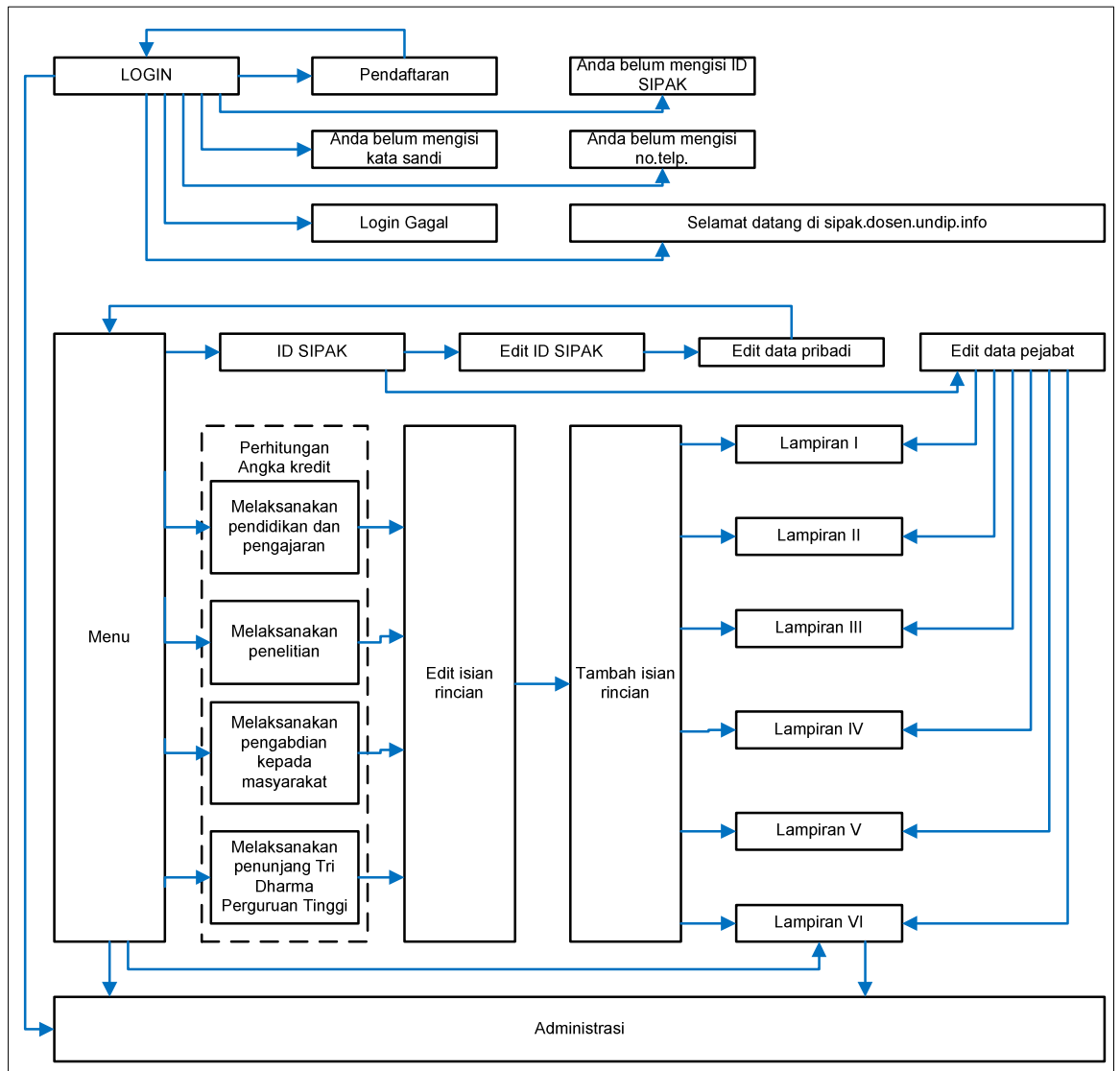
Aplikasi *Web* dibuat berdasarkan surat pernyataan melaksanakan kegiatan Tridharma Perguruan Tinggi (pendidikan dan pengajaran, penelitian, dan pengabdian pada masyarakat) dibuat menurut formulir sebagaimana tersebut pada lampiran sebagaimana tersebut pada Keputusan Menteri Negara Koordinator Bidang Pengawasan Pembangunan dan Pendayagunaan Aparatur Negara nomor : 38/kep/mk.waspan/8/1999.

### **3.6.3 Struktur Aplikasi Web**

Struktur aplikasi *web*, lihat gambar 3.4, terdiri dari Login, jika belum terdaftar wajib mendaftar terlebih dahulu di *sipak.dosen.undip.info*. Jika memaksakan diri akan ditolak “anda belum mengisi ID SIPAK”. Jika saat mendaftar tidak melengkapi data “login gagal”. Jika pengisian data kurang kata sandi, akan muncul web peringatan “anda belum mengisi kata sandi”. Jika nomor telepon tidak di isi, akan muncul web peringatan “anda belum mengisi nomor telepon”. Tetapi bilamana pendaftaran telah lengkap, akan disiruk untuk login

kembali, jika sesuai apa yang telah di isikan akan muncul selamat datang di *sipak.dosen.undip.info*.

Dalam halaman *web* selamat datang di *sipak.dosen.undip.info* terdiri menu yang berisi: beranda, data pribadi, angka kredit yang telah dimiliki, perhitungan angka kredit, print lampiran I, print lampiran II, print lampiran III, print lampiran IV, print lampiran V, print lampiran VI, dan keluar. Jika di klik “data pribadi” akan muncul *web* ID SIPAK, dalam ID SIPAK ini terdapat “edit ID SIPAK” jika sewaktu-waktu diganti, terdapat “edit data pribadi”, dan “edit data pejabat”. Edit data pribadi akan memperbaiki “data pribadi”, dan “edit data pejabat” akan memperbaiki data pejabat bilamana telah terjadi pergantian jabatan.



**Gambar 3.4** Struktur aplikasi web

Edit data pejabat ini akan masuk di web lampiran I sampai dengan web Lampiran VI. Data pribadi akan masuk ke data administrasi, tetapi harus dibuka lewat login terlebih dahulu karena mempunyai kata kunci tersendiri. Didalam menu terdapat “perhitungan angka kredit”, jika “perhitungan angka kredit” di klik akan muncul web penghitungan angka kredit yang terdiri dari: melaksanakan

pendidikan dan pengajaran, melaksanakan penelitian, melaksanakan pengabdian kepada masyarakat, dan melaksanakan penunjang Tri Dharma perguruan Tinggi. Masing-masing mempunyai *web* “edit isian rincian”, dari edit isian rincian ini akan masuk web “tambah isian rincian” terus ke *web* Lampiran I sampai dengan Lampiran VI. Dari menu data pribadi akan masuk ke *web* “administrasi”, dan dari Lampiran VI juga masuk ke web “administrasi” sebagai data pribadi dosen.

#### **3.6.4 Inputan Aplikasi Web**

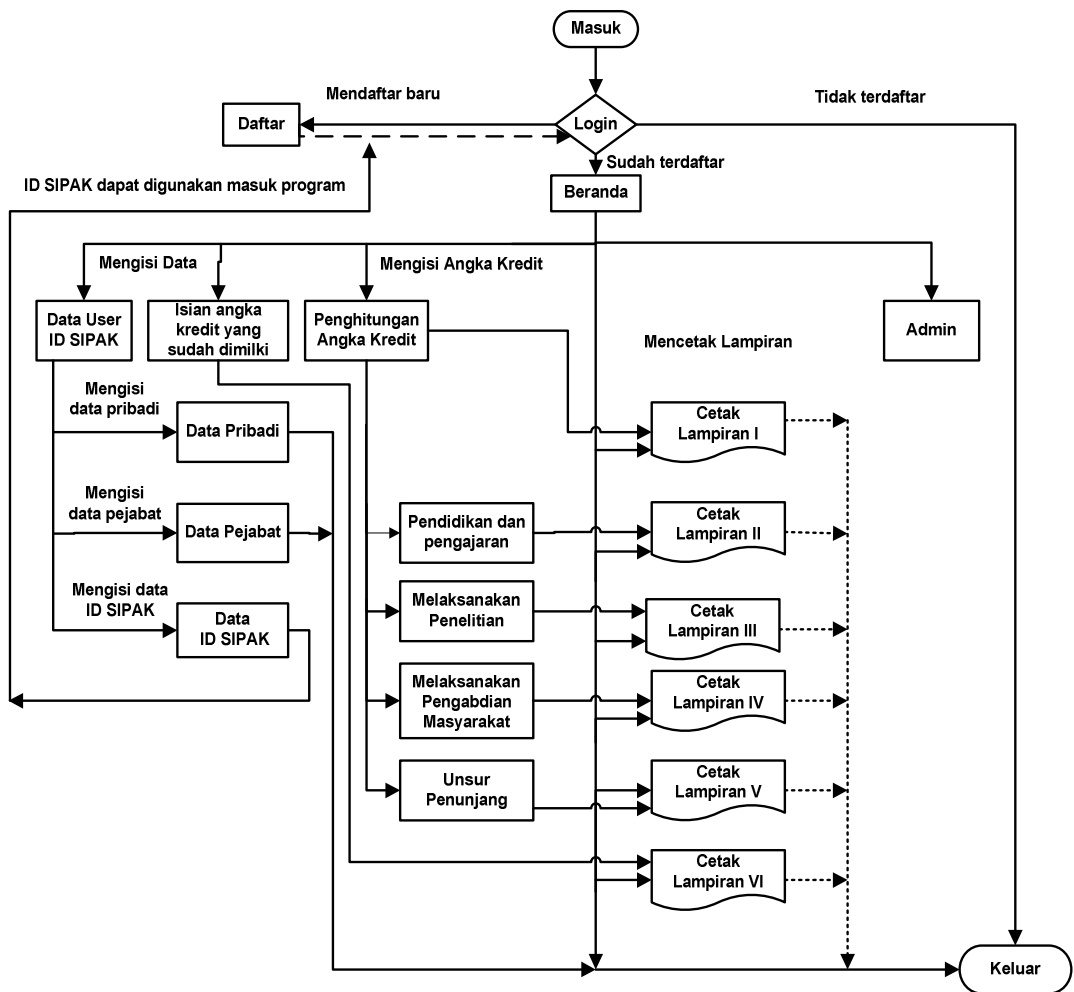
Angka kredit kinerja dosen merupakan inputan aplikasi *web*, inputan angka kredit dalam melaksanakan pendidikan dan pengajaran pada perguruan tinggi, penelitian serta pengabdian kepada masyarakat dan juga penunjang Tridharma Perguruan Tinggi merupakan data utama inputan aplikasi *web*.

Kinerja dosen dalam memperoleh angka kredit antara lain melalui:

- a) Pendidikan adalah pengembangan kemampuan dan jati diri peserta didik sebagai wujud kepribadian yang utuh, melalui program pengajaran yang diarahkan melalui kurikulum program studi.
- b) Pengajaran adalah pengembangan penalaran peserta didik untuk mendalami kaidah-kaidah keilmuan sebagai pelaksanaan tugas fungsional Dosen yang terdiri dari pemilihan dan pengorganisasian materi, pelaksanaan kegiatan pembelajaran, dan penilaian proses serta hasil pembelajaran sesuai dengan sasaran kurikulum yang telah ditentukan.
- c) Pendidikan akademik adalah pendidikan yang mengutamakan peningkatan penguasaan dan perluasan wawasan ilmu pengetahuan.

- d) Penelitian adalah kegiatan tela'ah taat kaidah dalam upaya untuk menemukan kebenaran dan/atau menyelesaikan masalah dalam ilmu pengetahuan, dan/atau kesenian.
- e) Karya ilmiah ialah karya yang mengikuti kaidah, peraturan dan jalan pikiran yang berlaku dalam ilmu pengetahuan serta memberikan sumbangan kepada khasanah ilmu pengetahuan di bidang masing-masing.
- f) Penulis utama satu karya ilmiah adalah penanggung jawab utama yang memprakarsai penulisan, pemilik ide tentang hal yang akan ditulis, pembuat kerangka, penyusun konsep, serta pembuat konsep akhir dari tulisan tersebut.
- g) Penulis pembantu adalah penulis lainnya diluar penulis utama.
- h) Angka kredit adalah satuan nilai dari tiap butir kegiatan dan atau akumulasi nilai butir-butir kegiatan yang diberikan/ditetapkan berdasarkan penilaian atas prestasi yang telah dicapai oleh seorang Dosen dan yang dipergunakan sebagai salah satu syarat dalam rangka pembinaan karier dalam jabatan fungsional/kepegangatan.

Pengguna aplikasi *web* agar dapat melakukan *entry data* lewat Internet atau Intranet harus melakukan login aplikasi *web* terlebih dahulu, setelah login akan muncul beranda selamat datang di web sistem informasi usulan angka kredit (SIPAK) Undip dot info. Dalam beranda SIPAK akan didapati menu utama, pengguna tinggal memelih menu utama yang dikendaki. Proses penggunaan aplikasi *web* seperti terlihat pada diagram alir gambar 3.6.



Gambar 3.5 Proses aplikasi web

### 3.6.5 Output Aplikasi Web

Aplikasi Web akan menghasilkan output berupa *printout hard copy* yang dapat dipergunakan untuk syarat pengajuan penetapan angka kredit jabatan fungsional dosen. Setelah pengguna, *user*, melakukan pengisian angka kredit sesuai dengan format yang telah ditentukan menurut aturan yang berlaku sebagaimana tersebut pada Keputusan Menteri Negara Koordinator Bidang

Pengawasan Pembangunan dan Pendayagunaan Aparatur Negara nomor: 38/kep/mk.waspan/8/1999.

### **3.7 Desain Database**

Dalam Wikipedia, dijelaskan bahwa desain *database* adalah proses menghasilkan model data rinci dari *database*. Model data logis berisi semua pilihan logis dan fisik penyimpanan parameter yang diperlukan untuk menghasilkan desain dalam *Data Definition Language*, yang kemudian dapat digunakan untuk membuat database. Sebuah model data sepenuhnya berisi atribut rinci untuk setiap entitas.

Desain database panjang dapat digunakan untuk menggambarkan berbagai belahan rancangan sistem basis data secara keseluruhan. Pada prinsipnya, *database* dapat dianggap sebagai desain logis dari struktur basis data yang digunakan untuk menyimpan data. Dalam model relasional ini adalah tabel dan tampilan. Dalam sebuah *database* objek entitas dan hubungan peta langsung ke objek kelas dan bernama hubungan. Namun, desain *database* juga dapat digunakan untuk keseluruhan proses perancangan, bukan hanya struktur *database*, tetapi juga bentuk dan *query* yang digunakan sebagai bagian dari aplikasi database secara keseluruhan dalam sistem manajemen *database* (DBMS).

Proses melakukan perancangan basis data umumnya terdiri dari sejumlah langkah yang akan dilakukan oleh *desainer database*. Biasanya, perancang harus

menentukan hubungan antara unsur-unsur data yang berbeda, dan menentukan struktur logis atas data berdasarkan hubungan ini.

### 3.7.1 Proses Desain *Database*

Proses desain terdiri dari langkah-langkah berikut:

- a) Menentukan tujuan dari *database*, ini membantu mempersiapkan untuk langkah berikutnya.
- b) Mencari dan mengatur informasi yang diperlukan. Mengumpulkan semua jenis informasi yang mungkin ingin direkam dalam database, seperti nama, nip, jabatan, pangkat, angka kredit, dan nomor ponsel.
- c) Membagi informasi ke dalam tabel. Bagi item informasi ke dalam entitas besar atau subjek.
- d) Item informasi masukan ke dalam kolom. Menentukan informasi apa yang ingin disimpan dalam setiap tabel. Setiap item menjadi lapangan, dan ditampilkan sebagai kolom dalam tabel. Misalnya, tabel dosen dapat mencakup bidang-bidang seperti *Last Name* dan NIP.
- e) Menentukan kunci primer. Primary key adalah kolom yang digunakan untuk secara unik mengidentifikasi setiap baris. Sebuah contoh mungkin NIP .
- f) Mengatur hubungan tabel, mlihat di setiap kolom dan memutuskan bagaimana data dalam satu tabel berhubungan dengan data dalam tabel

lain. Menambahkan *field* untuk tabel atau membuat tabel baru untuk memperjelas hubungan, bilamana diperlukan.

- g) Memperbaiki desain. Menganalisis desain jika ada kesalahan. Membuat tabel dan menambahkan beberapa catatan data sampel. Jika bisa mendapatkan hasil yang diinginkan dari tabel. Melakukan penyesuaian dengan desain sesuai kebutuhan.
- h) Menerapkan aturan normalisasi data untuk melihat apakah tabel terstruktur dengan benar. Membuat penyesuaian terhadap tabel.

Melakukan desain *database* adalah dengan keahlian di bidang desain *database*, bukan keahlian dalam domain dari mana data yang akan disimpan diambil misalnya informasi kepegawaian, informasi, dan lain lain. Demikian data yang akan disimpan dalam *database* harus ditentukan dengan bekerjasama dengan orang yang memang memiliki keahlian dalam domain tersebut, dan yang menyadari apa data harus disimpan dalam sistem.

Proses ini adalah salah satu yang umumnya dianggap sebagai bagian dari analisis kebutuhan, dan membutuhkan keterampilan pada bagian dari desainer *database* untuk memperoleh informasi yang dibutuhkan dari orang-orang dengan pengetahuan domain.

### 3.7.2 Normalisasi *Database*

Di bidang desain *database relasional*, normalisasi adalah cara sistematis untuk memastikan bahwa struktur *database* cocok untuk query tujuan umum dan bebas dari penyisipan tertentu yang tidak diinginkan-karakteristik, *update*, dan penghapusan *anomali*-yang bisa mengakibatkan hilangnya integritas data.

Standar pedoman desain *database* meripakan *desainer* harus membuat desain sepenuhnya normal, *denormalization* selektif selanjutnya dapat dilakukan, tapi hanya untuk alasan kinerja.

### **3.7.3 Skema Konseptual**

Seorang *desainer database* menyadari data yang akan disimpan dalam *database*, kemudian harus menentukan di mana ketergantungan berada dalam data. Kadang-kadang ketika data berubah dapat mengubah data lain yang tidak terlihat. Sebagai contoh, dalam daftar nama dan alamat, asumsi situasi di mana beberapa orang dapat memiliki alamat yang sama, tetapi satu orang tidak dapat memiliki lebih dari satu alamat, nama tergantung pada alamat, karena jika alamat yang berbeda, maka nama yang terkait pun berbeda.

Setelah hubungan dan ketergantungan antara berbagai potongan informasi yang telah ditentukan, adalah mungkin untuk mengatur data ke dalam suatu struktur logis yang kemudian dapat dipetakan ke dalam penyimpan benda-benda yang didukung oleh sistem manajemen *database*. Dalam kasus *database* relasional obyek penyimpanan tabel yang menyimpan data dalam baris dan kolom.

Dalam sebuah *database* objek penyimpan benda-benda sesuai langsung ke obyek yang digunakan oleh bahasa pemrograman berorientasi obyek yang digunakan untuk menulis aplikasi yang akan mengelola dan mengakses data. Hubungan dapat didefinisikan sebagai atribut dari kelas objek yang terlibat atau sebagai metode yang beroperasi pada kelas objek.

#### **3.7.4 Desain Database Fisik**

Desain fisik *database* menentukan konfigurasi fisik dari *database* pada media penyimpanan. Ini termasuk spesifikasi rinci elemen data, tipe data, opsi indeks dan parameter lain yang berada dalam data DBMS. Ini adalah desain rinci sistem yang mencakup modul, *hardware database*, dan spesifikasi perangkat lunak sistem.

Dalam dunia Internet, MySQL dijadikan sebuah database yang paling banyak digunakan (Nugroho B., 2007), penggunaan MySQL biasanya dipadu dengan program aplikasi PHP, karena dengan menggunakan kedua program tersebut keandalannya akan menjadi lebih baik.

MySQL merupakan sebuah bentuk *database* yang berjalan sebagai *server*, artinya tidak harus meletakkan *database* dalam satu mesin dengan aplikasi yang digunakan, sehingga dapat meletakkan sebuah *database* pada sebuah mesin khusus dan dapat diletakkan pada tempat yang jauh dari komputer pengaksesnya.

Sebagai bentuk *database server*, MySQL memiliki kemampuan untuk menangani *database* dengan menggunakan hak-hak sebagai user didalamnya, dengan adanya pembagian hak akses *user*, maka tidak semua anggota atau orang yang memiliki hak pada *server* tersebut tidak dapat dengan leluasa menggunakan atau mengakses *database* yang dimiliki orang lain atau *user* lain kecuali hanya yang dimiliki oleh *user* tersebut.

















































































































Dengan adanya hak *user* tersebut, *database* memerlukan adanya sebuah hak yang menjadi raja dalam sistem *database* tersebut, hak yang paling tinggi pada MySQL adalah *root*. *User root* akan menjadi user utama pada *database* ini, sehingga dapat melakukan pengaturan dalam sistem data base tersebut sampai dengan memberi hak-hak kepada *user* yang ada didalamnya.

Pada saat pertama kali MySQL dijalankan, *user* dapat masuk dengan akses penuh, tetapi tidak dapat melanjutkan tanpa mempunyai *password* yang telah tersimpan dalam *database*, maka *password* harus dibuat terlebih dahulu sebelum masuk *database* lebih dalam.

NIP dinyatakan sebagai *primary key* digunakan untuk *password* yaitu mengilustrasikan sebuah batasan terlebih dahulu, jika batasan dilanggar akan terjadi kendala dan dapat digunakan untuk mengidentifikasi kesalahan. Ramakrishnan, R; Gehrke, J. (2004, h. 49). Nama, *password*, sebagai *secondary key*, agar dapat *login* harus mengisikan *password*, jika *password* tidak diisi tidak dapat mengakses *web*.

Angka kredit diproses dan akan menghasilkan ”penetapan angka kredit” yang berguna untuk usulan naik jabatan fungsional. Masukan angka kredit berupa angka perolehan angka kredit dari tugas-tugas yang telah dijalankan, angka kredit pendidikan dan pengajaran, angka kredit penelitian, angka kredit pengabdian kepada masyarakat, dan angka kredit penunjang Tri Dharma Perguruan Tinggi. ”Penetapan angka kredit” merupakan kumulatif pencapaian hasil angka kredit yang telah diperoleh dari kinerja Dosen.

Recana *database* dosen dapat dilihat pada gambar 3.6.

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	username	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	password	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	nama_jengkap	varchar(100)	latin1_general_ci		No	None		      
<input type="checkbox"/>	nip	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	karpeg	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	ttd	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	kelamin	enum('Laki-laki','Wanita')	latin1_general_ci		No	Laki-laki		      
<input type="checkbox"/>	pend	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	penetapan	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	golong	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	id_golongan	int(5)			No	None		      
<input type="checkbox"/>	id_golongan2	int(2)			No	None		      
<input type="checkbox"/>	jabat	varchar(50)	latin1_general_ci		No	None		      
<input type="checkbox"/>	id_jabatan	int(5)			No	None		      
<input type="checkbox"/>	id_jabatan2	int(2)			No	None		      
<input type="checkbox"/>	fakultas	varchar(50)	latin1_general_ci		No	None		      

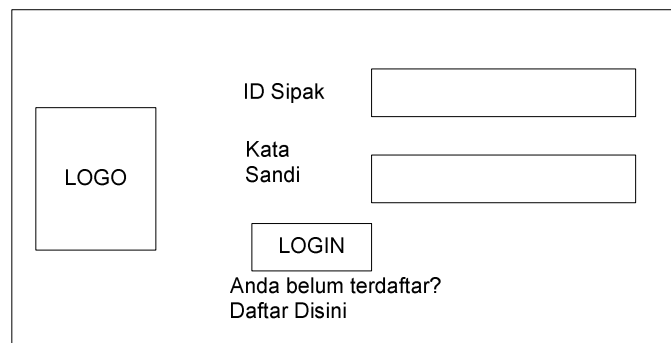
**Gambar 3.6** Rencana *database* administrasi

### 3.8 Desain Antarmuka

Antarmuka diperlukan untuk memudahkan komunikasi pengguna, *user*, dengan tampilan *web*, dengan kejelasan apa yang dimaksudkan sehingga tindak lanjut pengguna menyesuaikan tampilan *web* tersebut.

### 3.8.1 Web Antar Muka

Web antarmuka (*interface*) dirancang dengan mempertimbangkan kesederhanaan dan kejelasan desain agar mudah digunakan oleh user. Pertama kali untuk masuk ke web SIPAK harus LOGIN terlebih dahulu, tampilan LOGIN dirancang seperti gambar 3.7.

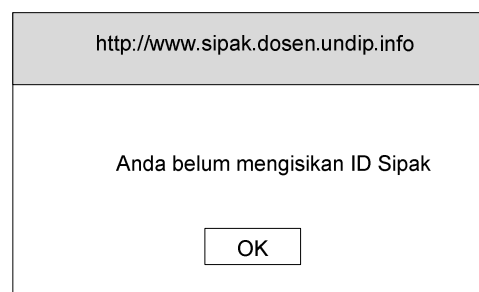


The image shows a login form with the following elements:

- A box labeled "LOGO" on the left side.
- Two input fields: "ID Sipak" and "Kata Sandi".
- A "LOGIN" button below the input fields.
- A link "Daftar Disini" below the login button, preceded by the text "Anda belum terdaftar?".

**Gambar 3.7** Desain *web* antarmuka LOGIN

Jika user belum mendaftar SIPAK tetapi memaksakan login, web SIPAK akan mengingatkan bahawa: **Anda belum mengisi ID Sipak**. Desain *web* mengingatkan user akan terlihat seperti gambar 3.8.



The image shows a dialog box with the following elements:

- A title bar with the URL "http://www.sipak.dosen.undip.info".
- Main text: "Anda belum mengisi ID Sipak".
- An "OK" button at the bottom.

**Gambar 3.8** Desain *web* mengingatkan user

User akan klik *button* OK dan *web link* ke LOGIN, user dengan klik **daftar disini** maka akan muncul *form web pendaftaran* dan user harus mengisi terlebih

dahulu agar *user* terdaftar di *web* SIPAK. Desain *web* pendaftaran dapat dilihat pada gambar 3.9.

PENDAFTARAN		
ID SIPAK	<input type="text"/>	Isilah ID SIPAK sama dengan no.NIP Anda
Kata Sandi	<input type="text"/>	Isilah kata sandi yang mudah diingat
Email	<input type="text"/>	Isilah Email Anda yang masih aktif
No.Telp/HP	<input type="text"/>	Isilah nomor telepon Anda yang masih aktif

**Gambar 3.9** Desain *web* pendaftaran

Setelah *user* mendaftarkan diri, berarti telah tercatat dalam *server web* SIPAK, untuk masuk ke dalam situs *web* SIPAK *user* harus *login* kembali dengan menuliskan ID Sipak dan *password* yang telah diisikan. Kalau *login* ini berhasil akan keluar beranda *web* SIPAK dengan desain seperti berikut.

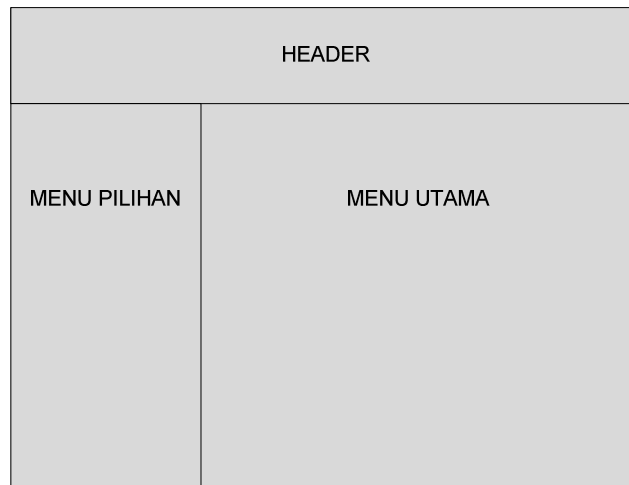
**“Selamat Datang di sipak.dosen.undip.info**

**APLIKASI WEB USULAN PENETAPAN ANGKA KREDIT  
JABATAN FUNSIONAL DOSEN**

**Dengan mengucapkan syukur kehadirat Allah SWT dengan terbitnya Sistem Informasi Usulan Penetapan Angka Kredit Jabatan Fungsional Dosen berbasis web dilingkungan Universitas Diponegoro. Diharapkan memudahkan Dosen dalam pengisian syarat-syarat pemenuhan angka kredit jabatan fungsional secara online dan memudahkan petugas kepegawaian memeriksa kinerja Dosen yang ingin naik jabatan fungsional.**

**Silakan klik menu pilihan yang berada disebelah kiri untuk pengisian data pribadi dan angka kredit”**

Tampilan halaman utama *web* dirancang terbagi menjadi tiga bagian, yaitu baris *header*, menu pilihan, dan bagian isi menu utama seperti gambar 3.10.



Gambar 3.10 **Desain web tampilan halaman utama**

Di bagian *header* ini berisi logo UNDIP, dan nama perangkat lunak sistem informasi. Desain *header* lebih detail seperti gambar gambar 3.11.



Gambar 3.11 **Desain header web**

Kolom menu pilihan di sisi kiri situs beranda, Isian Data Pribadi, Isian Angka Kredit yang telah diperoleh, Isian Angka Kredit, Lampiran I sampai dengan Lampiran VI, dan keluar. Bagian menu utama adalah *web* isian angka kredit yang merupakan bagian yang harus diisi oleh dosen setelah memperoleh angka kredit, di mana *user* (dosen) dalam mengeklik pada sistem *web* isian akan saling berhubungan dengan *web* lain dan memberikan input dan akan menghasilkan output.

### **3.8.2 Web Isian Data Pejabat**

*Web* data pejabat harus diisi karena mereka yang berwenang menandatangani syarat-syarat kenaikan jabatan fungsional dosen. Data pejabat yang diisikan di *web* data pejabbat ini akan otomatis mengisi data pejabat di Lampiran I sampai dengan Lampiran VI secara proporsional sesuai *link* yang dibuat untuk itu. Rancangan desain *web* isian data pejabat seperti terlihat pada gambar 3.12.

Golongan ruang dapat dipilih: Penata Muda (IIIa), Penata Muda Tk I (IIIb), Penata (IIIc), Penata Tk I (IIId), Pembina (IVa), Pembina Tk I (IVb), dan Pembina Utama Muda (IVc). Demikian halnya untuk jabatan fungsional dapat dipilih: Asiten Ahli, Lektor, Lektor Kepala, dan Guru Besar

HEADER	
MENU PILIHAN	<b>DATA PEJABAT</b>
	DEKAN <input type="text"/>
	Nama <input type="text"/>
	NIP <input type="text"/>
	Golongan Ruang <input type="text"/> ▼
	Jabatan Fungsional <input type="text"/> ▼ TMT <input type="text"/>
	Unit Kerja <input type="text"/>
	KETUA JURUSAN <input type="text"/>
	Nama <input type="text"/>
	NIP <input type="text"/>
	Golongan Ruang <input type="text"/> ▼
	Jabatan Fungsional <input type="text"/> ▼ TMT <input type="text"/>
	Unit Kerja <input type="text"/>
	Ketua Panitia Penilai Jabatan Tenaga Pengajar Pusat <input type="text"/>
NIP <input type="text"/>	

**Gambar 3.12** Desain *web* isian data pejabat

### 3.8.3 *Web* Isian Data Pribadi

Desain *web* data pribadi, lihat gambar 3.13, diisikan oleh user (dosen) secara otomatis akan mengisi *database* administari yang nantinya hanya dapat di akses oleh petugas administrasi. Petugas admistrasi dapat langsung melihat perolehan angka kredit dosen menurut hasil kinerjanya.

HEADER		
MENU PILIHAN	RINCIAN DATA PRIBADI	
	<b>DATA PRIBADI</b>	
	Nama	<input type="text"/>
	NIP	<input type="text"/>
	No. Seri Karpeg	<input type="text"/>
	Tempat/tanggal lahir	<input type="text"/>
	Jenis Kelamin	<input type="radio"/> Laki-laki <input type="radio"/> Wanita
	Pendidikan tertinggi	<input type="text"/>
	Golongan Ruang	<input type="text" value="▼"/> TMT <input type="text"/>
	Jabatan Fungsional	<input type="text" value="▼"/> TMT <input type="text"/>
	Fakultas/Jurusan	<input type="text"/>
	Impasing	<input type="text"/>
	Masa Kerja	Lama <input type="text"/>
		Baru <input type="text"/>
	Unit Kerja	<input type="text"/>
	Tanggal penetapan	<input type="text"/>
	<b>DIUSULKAN</b>	
	Golongan Ruang	<input type="text" value="▼"/>
	Jabatan Fungsional	<input type="text" value="▼"/>
	<input type="button" value="UPDATE"/> <input type="button" value="BATAL"/>	

**Gambar 3.13** Desain *web* isian data pribadi

Golongan ruang dapat dipilih dari Penata Muda (IIIa), Penata Muda Tk I (IIIb), Penata (IIIc), Penata Tk I (IIId), Pembina (IVa), Pembina Tk I (IVb), dan Pembina Utama Muda (IVc). Demikian halnya nuntuk jabatan fungsional dapat dipilih Asiten Ahli, Lektor, Lektor Kepala, dan Guru Besar dan diakhiri klik *button* UPDATE.

### 3.8.4 *Web* Isian Perhitungan Angka Kredit

*Web* isian perhitungan angka kredit adalah isian Lampiran I, dalam poin isian angka kredit ini terdapat button Edit, jika *button* Edit di klik akan muncul isian *web* pembantu isian rincian, jika *web* isian rincian ini diisi secara otomatis akan mengisi *web* Lampiran I sampai dengan *web* Lampiran VI.

Menurut Keputusan Menteri Negara Koordinator Bidang Pengawasan Pembangunan dan Pendayagunaan Aparatur Negara nomor : 38/kep/mk.waspan/8/1999 yang ditampilkan menjadi *web applications*. *Web* inilah media pengisian Sistem Informasi Penetapan Angka Kredit (SIPAK) jabatan fungsional dosen. *Web* isian dirancang sedemikian sederhana supaya *user* (dosen) mudah dalam melakukan pengisiannya.

Desain *web* isian perhitungan angka kredit dapat dilihat pada gambar 3.14.

HEADER			
Perhitungan Angka Kredit			
NO	UNSUR YANG DINILAI	ANGKA KREDIT	AKSI
I	UNSUR UTAMA		
A	PENDIDIKAN		
a	Mengikuti pendidikan sekolah dan memperoleh gelar/sebutan/ijazah/akta		Edit
b	Mengikuti pendidikan sekolah dan memperoleh gelar/sebutan/ijazah/akta diluar bidang ilmunya		Edit
c	Mengikuti pendidikan dan pelatihan fungsional dosen dan memperoleh STTPL		Edit
J U M L A H			
B	TRI DHARMA PERGURUAN TINGGI		
a	MELAKSANAKAN PENDIDIKAN DAN PENGAJARAN		Edit
(1)	Melaksanakan perkuliahan/tutorial dan membimbing mengji serta menyelenggarakan pend di lab		Edit
(2)	Membimbing seminar mahasiswa		Edit
(3)	Membimbing KKN, PKN, PKL		Edit
(4)	Membimbing tugas akhir/skripsi/tesis/disertasi		Edit
(5)			Edit
(12)	Melaksanakan kegiatan detasiring/pencangkokan dosen		Edit
J U M L A H			
b	MELAKSANAKAN PENELEITIAN		
(1)	Menghasilkan karya ilmiah		Edit
(2)	Menterjemahkan/menyadur bulu ilmiah		Edit
(3)	Mengedit/menyunting karya ilmiah		Edit
(4)	Membuat rancangan dan karya tehnologi yang dipatenkan		Edit
(5)	Membuat rancangan dan karya tehnologi rancangan dan karya seni monumental		Edit
J U M L A H			
c	MELAKSANAKAN PENGABDIAN PADA MASYARAKAT		
(1)	Menduduki jabatan pimpinan pada lembaga pemerintah/pejabat negara		Edit
(2)	Melaksanakan pengembangan hasil pendidikan dan penelitian yang dapat dimanfaatkan masy		Edit
(3)	Memberikan latihan/penyuluhan/penataran/ceramah pada masyarakat		Edit
(4)	Memberikan pelayanan kepada masyarakat atau kegiatan lain yang menunjang pembangunan		Edit
(5)	Membuat/menulis karya pengabdian pada masyarakat yang tidak dipublikasikan		Edit
J U M L A H			
2	UNSUR ENUNJANG		
d	PENUNJANG TUGAS PKOK DOSEN		
(1)	Menjadi anggota dalam suatu panitia/badan pada Perguruan Tinggi		Edit
(2)			Edit
(9)	Mempunyai prestasi di bidang olah raga/humaniora		Edit
J U M L A H			

MENU PILIHAN

Klik disini

UPDATE

BATAL

**Gambar 3.14** Desain web perhitungan angka kredit

Jika **button Edit** di klik akan terbuka *web* Edit isian rincian seperti gambar

3.15.

The image shows a web form titled "Edit Isian Rincian" with a "HEADER" section. On the left, there is a "MENU PILIHAN" sidebar. The main form area contains the following fields and controls:

- Kode\_Keg:
- Butir Kegiatan:
- Isian kegiatan:
- Tempat/Instansi:
- Semester/Tanggal:
- Jumlah Angka Kredit:
- Keterangan/Bukti Fisik:
- Aktif:  Y  N
- Buttons:

A callout box with a pointer to the "Jumlah Angka Kredit" field contains the text: "Ke Lampiran I dan ke Admin".

**Gambar 3.15** Desain *web* edit isian rincian angka kredit

Jumlah angka kredit yang diisikan akan otomatis menjumlah di *web* Lampiran I, di *web* Lampiran II, dan di *web* Lampiran VI, juga di *web* Administrasi. Hal ini akan terjadi jika **Aktif** di klik **Y**. **Button Update** berfungsi untuk memperbaiki data yang telah diperoleh, **button Batal** berfungsi jika akan membatalkannya.

### 3.8.5 Web Administrasi

*Web* administrasi merupakan data kumpulan data pribadi yang telah mengisi isian data pribadi, data pribadi meliputi: nama, nip, pangkat dan golongan

ruang saat ini, jabatan fungsional saat ini, diusulkan pangkat dan golongan, diusulkan jabatan fungsional, angka kredit, nomor ponsel, dan aksi. Aksi terdiri dari **Edit** dan **Warna**, Warna disini untuk membedakan dosen yang telah memenuhi angka kredit berwarna biru dan dosen yang belum memenuhi angka kredit berwarna merah. Perbedaan warna merah dan biru ini akan memudahkan petugas administrasi dalam melihat dosen yang telah memenuhi angka kredit dan yang belum memenuhi angka kredit, yang telah memenuhi angka kredit dapat segera dihubungi lewat nomor ponsel yang dicantumkan. Desain *web* data pribadi dan administrasi dapat dilihat pada gambar 3.16.

DATA PRIBADI				HEADER						
MENU PILIHAN	NAMA	NIP	PANGKAT/GOL. RUANG	JABATAN FUNGSIONAL	PANGKAT/GOL. RUANG	JABATAN FUNGSIONAL	ANGKA KREDIT	NO PONSEL	AKSI	
			SAAT INI		DIUSULKAN				EDIT	WARNA

**Gambar 3.16** Desain *web* data pribadi dan administrasi

Pemenuhan angka kredit tergantung kondisi saat ini mempunyai jabatan fungsional apa dan akan diusulkan ke jabatan fungsional yang lebih tinggi. Angka kredit fungsional jabatan dosen sesuai dengan Keputusan Menteri Negara Koordinator Pengawas Pembangunan dan Pendayagunaan Aparatur Negara Nomor: 38/Kep/MK.WASPAN/ 8/1999, Tanggal: 24 Agustus 1999. Lihat Lampiran H.

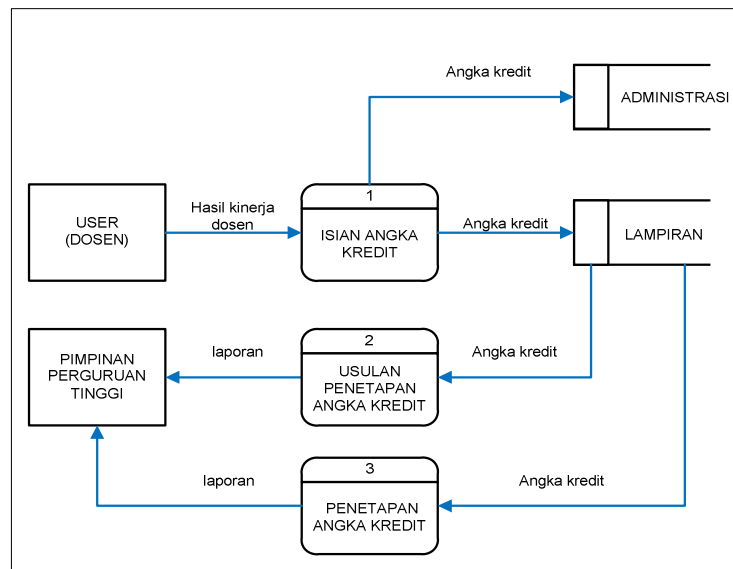
### **3.9 Perancangan Sistem Usulan Penetapan Angka Kredit**

Perancangan sistem akan memaparkan rancangan-rancangan sistem yang akan dibangun dengan menggunakan alat bantu diagram alir data (DFD).

#### **3.9.1 Diagram Alir Data (*Data Flow Diagram - DFD*) Angka Kredit**

Simbol diagram alir data yang digunakan adalah simbol menurut Tabel 2.1. Diagram alir data dalam perancangan ini akan terdiri dari DFD level 1, dan DFD level 2. Dalam DFD level 1, gambar 3.17.

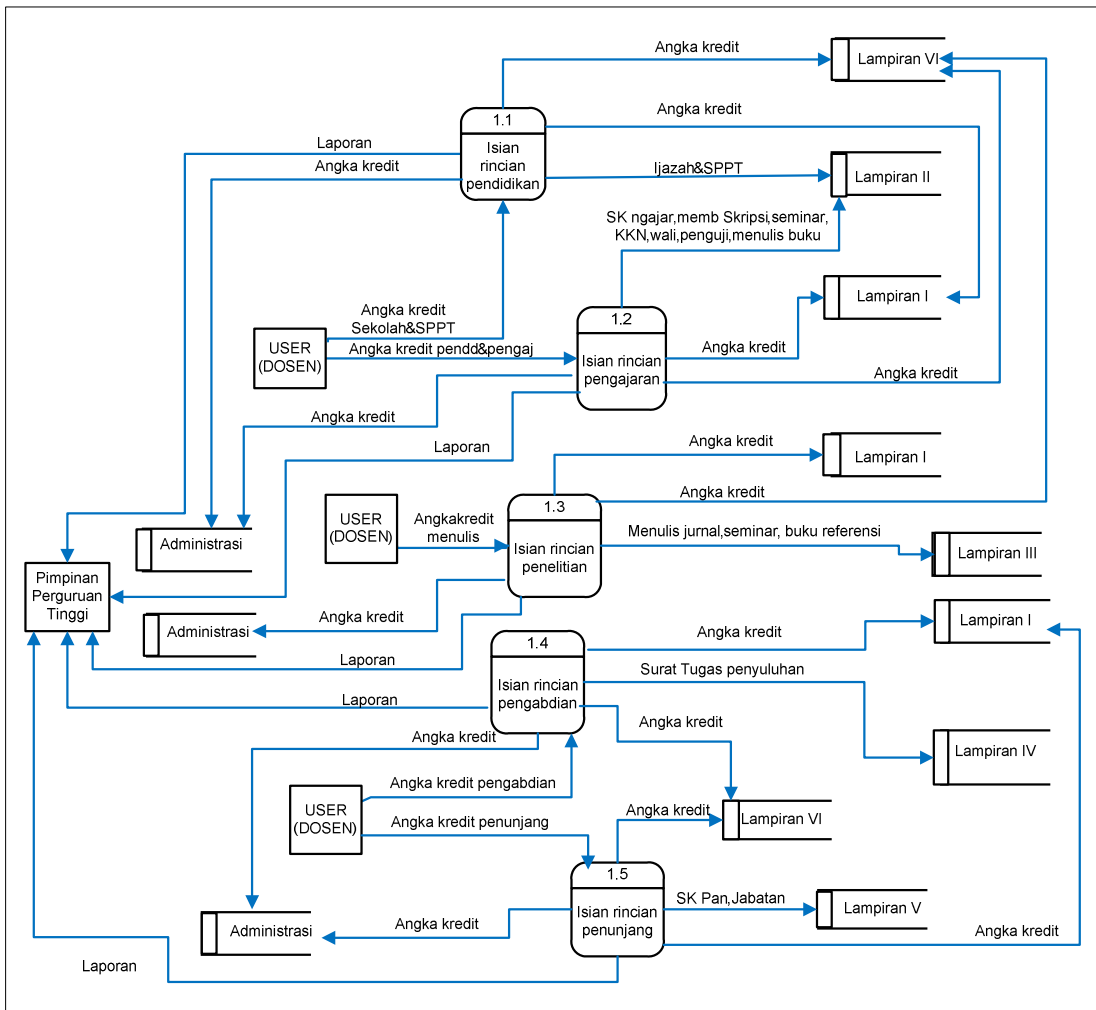
Pertama-tama *user* (dosen) akan mendaftar terlebih dahulu agar dicatat dalam sistem *web* SIPAK. Kemudian sistem informasi akan menyimpan data pribadi dosen, setelah dicatat dalam SIPAK, *user* harus melakukan login kembali untuk menampilkan *web* utama. Dalam *web* utama *user* (dosen) dapat memilih menu utama, dosen akan mengisikan hasil kinerjanya dalam “isian angka kredit” yang akan masuk ke *web* lampiran. Dalam *web* lampiran ada ada *web* “usulan angka kredit” dan *web* “penetapan angka kredit” yang nanti sebagai “laporan” untuk diusulkan ke “Pimpinan Perguruan Tinggi”. Saat melakukan isian angka kredit, angka kredit juga akan masuk ke dalam *web* Administrasi.



**Gambar 3.17** DFD Level 1 isian angka kredit

Dalam DFD Level 2, gambar 3.18, terdiri dari DFD level 1.1, DFD level 1.2, DFD level 1.3, DFD level 1.4, dan DFD level 1.5., menjelaskan bahwa *user* (dosen) yang melakukan pengisian angka kredit diproses dan menghasilkan laporan dari berbagai kinerjanya, dari pendidikan sekolah, pendidikan pengajaran, penelitian, pengabdian pada masyarakat, dan penunjang, untuk dilaporkan kepada pimpinan Perguruan Perguruan Tinggi.

Dalam *web* utama user dapat memilih menu utama, antara lain: Isian Angka Kredit yang telah diperoleh, Isian Angka Kredit, Lampiran I, Lampiran II, Lampiran III, Lampiran IV, Lampiran V, Lampiran VI, dan Data Pribadi. Saat melakukan isian angka kredit, angka kredit juga akan masuk ke dalam *web* Administrasi.



**Gambar 3.18** DFD level 2 rincian isian angka kredit

### 3.9.2 DFD Angka Kredit Pendidikan Sekolah

Bilamana user dalam *web* Penghitungan Angka kredit meng-klik Edit di Pendidikan Sekolah, angka kredit pendidikan sekolah masuk dalam DFD Level 1.1, ini merupakan proses rincian isian angka kredit pendidikan sekolah yang diperoleh setelah mengikuti pendidikan sekolah dan memperoleh gelar/sebutan/ijazah/akta sesuai bidang ilmunya, memperoleh gelar/sebutan/ijazah/akta tambahan yang setingkat atau lebih tinggi diluar bidang

ilmunya. Dan juga memperoleh angka kredit setelah mengikuti pendidikan dan pelatihan fungsional dan memperoleh Surat Tanda Tamat Pendidikan dan Pelatihan (STTPL). Angka-angka kredit yang diperoleh dalam kegiatan tersebut akan masuk kedalam Lampiran II, dan masuk Lampiran I, dan masuk ke Lampiran VI. Proses rincian angka kredit terus ke *web* Administrasi dan juga akan menjadi laporan ke pimpinan perguruan tinggi.

### **3.9.3 DFD Angka Kredit Pendidikan dan Pengajaran**

Bilamana user dalam *web* Penghitungan Angka kredit meng-klik Edit Pendidikan dan pengajaran akan masuk pada DFD Level 1.2, proses rincian isian pendidikan dan pengajaran meliputi dua belas kegiatan, yaitu: 1) Melaksanakan perkuliahan/tutorial dan membimbing, menguji serta menyelenggarakan pendidikan di laboratorium, praktek keguruan, bengkel/studio/kebun percobaan/teknologi pengajaran dan praktik lapangan; 2) Membimbing seminar mahasiswa; 3) Membimbing kuliah kerja nyata (KKN), praktek kerja nyata (PKN), praktik kerja lapangan (PKL); 4) Membimbing dan ikut membimbing dalam menghasilkan laporan akhir studi/skripsi/thesis/disertasi; 5) Bertugas sebagai npenguji pada ujian akhir; 6) Membina kegiatan mahasiswa di bidang akademik dan kemahasiswaan; 7) Mengembangkan program kuliah; 8) Mengembangkan bahan pengajaran; 9) Menyampaikan orasi ilmiah; 10) Menduduki jabatan pimpinan perguruan tinggi; 11) Membimbing dosen yang lebih rendah jabatan fungsionalnya; dan 12) Melaksanakan kegiatan detasering dan pencakokan dosen. Tetapi dalam gambar 3.18 hanya ditulis sebagian dan disingkat mengingat tempat

terbatas. Nilai angka kredit tersebut akan masuk ke dalam Lampiran II, masuk web Lampiran I, dan masuk ke web Lampiran VI. Rincian isian angka kredit juga akan masuk ke web Administrasi serta akan menjadi laporan ke pimpinan perguruan tinggi.

#### **3.9.4 DFD Angka Kredit Penelitian**

Bilamana user dalam Penghitungan Angka Kredit meng-klik meng-klik Edit Penelitian, proses memasukkan angka kredit masuk di DFD Level 1.3, proses rincian isian penelitian meliputi angka kredit: karya ilmiah, menterjemahkan/menyadur buku ilmiah, mengedit/menyunting karya ilmiah, membuat rancangan dan karya teknologi yang dipatenkan, dan membuat rancangan dan karya teknologi, rancangan dan karya seni monumental/seni pertunjukkan/karya sastra. Angka kredit yang dimasukkan ke proses rincian angka kredit penelitian akan masuk ke Lampiran III, masuk ke web Lampiran I dan web Lampiran VI. Proses rincian angka kredit terus ke web Administrasi dan menjadi laporan ke pimpinan perguruan tinggi.

#### **3.9.5 DFD Angka Kredit Pengabdian pada Masyarakat**

Bilamana user dalam Penghitungan Angka Kredit meng-klik Edit Pengabdian pada Masyarakat, proses memasukkan nilai angka kredit pengabdian pada masyarakat di DFD Level 1.4, proses rincian isian angka kreditnya meliputi: menduduki jabatan pimpinan pada lembaga pemerintah/pejabat Negara yang harus dibebaskan dari jabatan organiknya, melaksanakan pengembangan hasil pendidikan

dan penelitian yang dapat dimanfaatkan oleh masyarakat, memberi latihan/penyuluhan/penataran/ceramah pada masyarakat, member pelayanan kepada masyarakat atau kegiatan lain yang menunjang pelaksanaan tugas umum pemerintahan dan pembangunan, dan membuat/menulis karya pengabdian pada masyarakat. Proses pemasukan nilai angka kredit ini akan masuk ke proses rincian pengabdian pada masyarakat masuk *web* ke Lampiran IV, masuk ke masuk ke web Lampiran I dan web Lampiran VI. Proses rincian angka kredit terus ke web Administrasi dan menjadi laporan ke pimpinan perguruan tinggi.

### **3.9.6 DFD Angka Kredit Penunjang Tugas Pokok Dosen**

Bimana user dalam Penghitungan Angka Kredit meng-klik Edit Penunjang, proses memasukkan nilai angka kredit melakukan kegiatan penunjang tugas pokok dosen masuk pada DFD Level 1.5, proses isian rincian meliputi sembilan kegiatan:

- 1) Menjadi anggota dalam suatu panitia/badan pada perguruan tinggi;
- 2) menjadi anggota panitia/badan pada lembaga pemerintah;
- 3) Menjadi anggota organisasi profesi;
- 4) Mewakili perguruan tinggi/lembaga pemerintah duduk dalam panitia antar lembaga;
- 5) Menjadi anggota delegasi nasional ke pertemuan internasional;
- 6) Berperan serta aktif dalam pertemuan ilmiah;
- 7) Mendapat tanda jasa/penghargaan;
- 8) Menulis buku pelajaran SLTA ke bawah yang diterbitkan dan diedarkan secara nasional; dan
- 9) Mempunyai prestasi di bidang olah raga/humaniora. Nilai angka kredit tersebut masuk dalam proses rincian isian angka kredit penunjang masuk ke Lampiran V, terus ke masuk ke masuk ke web

Lampiran I dan web Lampiran VI. Proses rincian angka kredit terus ke web Administrasi dan menjadi laporan ke pimpinan perguruan tinggi.