**2. TINJAUAN PUSTAKA**

1. **Sistem *Presensi***

Sistem merupakan bagian-bagian atau prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sistem adalah kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (input) yang ditujukan kepada sistem tersebut dan mengubah masukan tersebut sampai menghasilkan keluaran (output) yang diinginkan. (Kristanto, 2007)

Presensi adalah pencatatan dan pengolahan data presensi yang dilakukan secara terus-menerus. Pencatatan dilakukan setiap hari kerja dan dilakukan pelaporan. Presensi pegawai merupakan salah satu tolak ukur metode pengembangan pegawai. Jika presensi pegawai setelah mengikuti pengembangan meningkat, maka metode pengembangan yang dilakukan baik, sebaliknya jika presensi pegawai tetap berarti metode pengembangan yang diterapkan kurang baik.

Program aplikasi presensi yaitu suatu aplikasi yang bergerak di bidang proses pendataan karyawan, yang terdiri dari penginputan data karyawan, data presensi karyawan dan pencetakan laporan. Pendataan karyawan bertujuan untuk memasukkan data karyawan. Data presensi karyawan bertujuan untuk memasukkan data presensi karyawan yang dasarnya menyatakan karyawan itu bekerja pada hari saat dia bekerja atau tidak.

1. **Agen Lapangan**

Bagi perusahaan yang bergerak di bidang jasa, keberadaan agen-agen yang turun ke lapangan sangatlah penting. Agen lapangan ini yang akan menjadi kekuatan perusahaan dalam memaksimalkan pendapatan dengan turun langsung ke lapangan dan menghadapi masalah yang ada di lingkungan pekerjaannya.

1. ***Web***

Ada beberapa ahli yang mengemukakakan beberapa pengertian dan juga definisi mendasar mengenai apa itu sebuah *web*. Berikut ini adalah beberapa pengertian *web* menurut para ahli:

1. Suwanto Raharjo

Menurut Suwanto Raharjo, *Web* merupakan salah satu layanan internet yang paling banyak digunakan oleh *user*nya dibandingkan dengan layanan lain seperti, seperti layanan internet *ftp*, *gopher****,*** *news* dan juga layanan surat elektronik atau *email*.

1. Wahana Komputer

Menurut Wahana Komputer, *Web* merupakan sebuah formulir komunikasi interaktif yang digunakan pada suatu jaringan komputer, salah satunya adalah penggunaan pada jaringan internet.

1. Taufiq Hidayatullah

Menurut A. Taufiq Hidayatullah, *Web* merupakan bagian dari sebuah aplikasi di dalam internet paling terlihat, dan juga paling banyak digunakan oleh *user* atau *client*nya.

1. Haertalib

Menurut Haertalib, *Web* merupakan sebuah lokasi di dalam jaringan internet, dimana lokasi tersebut memiliki tempat dan juga alamt tertentu, sehingga bisa diakses siapa saja.

1. Boone

Menurut Boone, *Web* merupakan sebuah koleksi sumber informasi yang kaya akan desan dan juga grafis yang mana dapat saling berhubungan satu sama lain dalam suatu jaringan internet yang lebih besar.

1. Feri Indayudha

Menurut Feri Indayudha, *Web* adalah suatu program yang dapat memuat banyak hal, seperti film, gambar, suara, serta musik yang dapat ditampilkan dan juga dilihat, dengan menggunakan teknologi jaringan internet yang ada.

1. Yuhefizar

Menurut Yuhefizar, *Web* merupakan suatu metode untuk menampilkan informasi di dalam jaringan internet, baik berupa teks, gambar, suara, maupun video yang interaktif. Lebih lanjut disebutkan pula bahwa *web* mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*.

Dari penjabaran diatas, maka dapat dikatakan bahwa yang dimaksud dengan *web* merupakan suatu metode menampilkan informasi di dalam *browser.* Penampilan informasi ini membutuhkan *user* untuk melakukan *request*, dan juga *software* yang digunakan untuk menampilkan informasi yang direquest tersebut. *Software* yang biasa digunakan, seperti *web* *browser* yang biasa kita gunakan. *(www.dosenit.com)*

1. ***Database***

*Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur file yang menyusun sebuah *database* adalah sebgai berikut:

1. Data

Data adalah satu satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan di dalam suatu file *database*.

1. *Record*

*Record* adalah data yang isinya merupakan satu kesatuan seperti Nama*User* dan *Password*. Setiap keterangan yang mencakup Nama*User* dan *Password* dinamakan satu *record*. Setiap *record* diberi nomor urut yang disebut nomor *record* (*Record* Number).

1. *Field*

*Field* adalah sub bagian dari *record*. Dari contoh isi *record* di atas, maka terdiri dari 2 *field*, yaitu: *Field* Nama*User* dan *Password*. *(Anhar-2010)*

Dalam pemrosesan nya, *database* memiliki berbagai macam perkembangan dari periode ke periode. Berikut rangkuman sejarah pemrosesan *database* yang disajikan pada tabel 2.1

Tabel 2.1 Rangkuman Sejarah Pemrosesan *Database*

|  |  |  |  |
| --- | --- | --- | --- |
| **Timeframe** | **Teknologi** | **Remarks** | |
| Sebelum -1968 | Pemrosesan file | Pendahulu pemrosesan database. Data disimpan dalam daftar. Karakteristik pemrosesan ditentukan oleh penggunaan umum media pita magnetik. | |
| 1968 – 1980 | Hierarkis dan model network | Era pemrosesan *database* non-relasional. Model data hierarkis yang terkemuka adalah DL/I, yaitu versi pertama DBMS IBM yang disebut IMS. Model data network terkemuka adalah model CODASYL DBTG. IDMS network yang paling populer. | |
| 1980 - hingga kini | Model data relasional | Model data relasional, yang dipublikasikan pertama kali pada tahun 1970. Di aplikasikan secara komersial pada tahun 1980. IBM menyebutkan dengan DB2, vendor lainnya mengikuti dengan memodifikasi produk DBMS-nya atau dengan menciptakan produk baru. Oracle mencapai puncaknya, SQL menjadi bahasa relasional standar. | |
| 1982 | Produik DBMS mikrokomputer pertama | Ashton-Tate mengembangkan produk dBase. Microrim menciptakan R:Base. Borland membuat Paradox. | |
| **Timeframe** | **Teknologi** | **Remarks** | |
| 1985 | Berkepentingan dalam pengembangan DBMS (OODBMS) yang berorientasi objek | Dengan ditemukannya pemrograman berorientasi objek, diusulkan OODBMS. Produk ini hanya meraih kesuksesan yang kecil secara komersial, terutama karena keunggulannya tidak menjustifikasi biaya mengkonversi miliaran byte data organisasi ke format baru. Saat ini masih dalam tahap pengembangan. | |
| 1991 | Microsoft ships Access | DBMS personal diciptakan sebagai unsu Windows. Secara bertahap menggantikan semua produk DBMS personal lainnya. | |
| 1995 | Aplikasi pertama *database* Internet | *Database* menjadi komponen kunci dari aplikasi Internet. Popularitas Internet meningkatkan kebutuhan dan permintaan akan keahlian *database*. | |
| 1997 | Penerapan XML | | Penggunaan XML memecahka masalah pemahaman *database* jangka panjang. Vendor utama mulai mengintegrasi XML ke produk DBMS |

Dengan suksesnya media penyimpan disk pada tahun 1960-an, kita dapat memiliki akses non sekuensial, atau langsung, ke *record*s. Dalam hal ini, *database* dirancang untuk menghilangkan masalah pemrosesan file sekuensial. Ada dua arsitektur atau model yang awalnya sukses. IBM mengembangkan dan mempromosikan DL/I atau *Data Language One*, yang membuat model *database* dalam bentuk pohon atau hirarki. Model ini, yang dikembangkan dalam kaitannya dengan industri manuaktur, mudah digunakan untuk menyimpan data seperti daftar *bill of material* dan suku cadang, tetapi sebenarnya bukan untuk tujuan umum. Menyajikan data network non-hierarkis ternyata cukup sulit.

Karena itu, CODASYL, kelompok yang mengembangkan standar untuk bahasa COBOL menciptakan sebuah model DBTG (*Data Base Task Group*) padat tahun 1970-an. Model DBTG dapat mewakili sistem hierarki dan *network.* Model ini pernah diajukan sebagai standar nasional, tetapi tidak pernah dipilih karena rumit. Akan tetapi, model ini telah menjadi dasar dari sejumlah produk DBMS yang sukses pada tahun 1970an dan 1980an. Produk IDMS buatan Cullinane Coorporation adalah yang paling sukses. *(Kroenke, David M, 2015)*

* 1. **MySQL**

MySQL adalah sebuah sistem manajemen *database* relasi (*relational* *database* *management system*) yang bersifat *open source*. (*Arbie, 2004*) MySQL merupakan buah pikiran dari Michael “Monty” Widenius, David Axmark dan Allan Larson yang di mulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di Swedia.

Software database MYSQL kini dilepas sebagai software manajemen database yang *open source,* sebelumya merupakan software database yang *shareware. Database* MYSQL tersedia secara bebas cuma-cuma dan boleh digunakan oleh setiap orang, dengan lisensi *open source* GNU *General Public Lisence* (GPL) ataupun lisensi komersil non GPL.

* 1. **HTML**

*HTML* adalah kepanjangan dari *HyperText Markup Language*, merupakan bahasa interpretasi yang digunakan pada sebuah halaman *web*. *HTML* mendeskripsikan struktur halaman *web* yang ditulis dengan elemen atau *tag* yang yang mengapit konten atau teks didalamnya.

Penjelasan lebih rinci mengenai arti kata-perkata dari *HTML* adalah sebagai berikut:

1. *HyperText*: adalah istilah teks aktif, yang apabila diklik akan meloncat atau menuju halaman lain. Ini merupakan kemampuan dari sebuah halaman *web* yang dapat saling berhubungan antara halaman satu dengan lainnya.
2. *Markup*: Merupakan *tag*-*tag* yang biasanya diawali dengan *tag* pembuka (*opening* *tag*) dan *tag* penutup (*closing* *tag*) yang memberi kemampuan untuk menata *layout* atau memformat struktur halaman *web* pada sebuah konten teks sederhana didalam *file* *HTML* itu sendiri.
3. *Language*: yaitu bahasa yang digunakan oleh *HTML* itu sendiri. Perintah-perintah *tag* menggunakan bahasa yang dapat dimengerti oleh *browser* atau *interpreter* lainnya.

*HTML* bukanlah sebuah bahasa pemrograman pada umumnya, seperti Java, C, C++, Visual Basic dan sejenisnya, melainkan bahasa *markup* / markah yang ditulis dengan perintah *tag*-*tag* atau elemen yang mengapit konten didalamnya yang akan ditampilkan pada sebuah halaman *web* oleh *browser* atau *HTML* *interpreter* (penerjemah *HTML*) lainnya.

*HTML* berguna untuk menampilkan konten, menghubungkan (*link*) antar halaman, memberi struktur dan informasi terkait dengan sebuah halaman *web*. Konten sebuah *web* tidak hanya terbatas pada teks saja, melainkan konten interaktif lainnya seperti video, audio, gambar dan animasi dapat disisipkan dan ditampilkan pada halaman *web*. *(Betha Sidik, Pohan, Husni Iskandar, 2014)*

* 1. **CSS**

CSS merupakan singkatan dari *cascade style sheet*, merupakan fitur baru dari HTML 4.0. hal ini diperlukan setelah melihat perkembangan HTML menjadi kurang praktis karena *web* *pages* terlalu banyak dibebani, hal-hal ini yang berkaitan dengan faktor tampilan seperti *font* dan lain-lain.

Untuk itu kumpulan *style* dikelola secara terpisah maka manajemen *pages* menjadi lebih mudah dan efisien, pada prakteknya penggunaan CSS ini didukung oleh *Explorer* dan *Navigator* browser terpopuler pada internet. *(Betha Sidik, Pohan, Husni Iskandar, 2014)*

* 1. **Backend**
  2. **PHP**

PHP merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembangan *web*. PHP pertama kali dikembangkan oleh Rasmus Lerdorf, seorang pengembang *software* dan anggota tim Apache, dan dirilis pada akhir tahun 1994. PHP dikembangkan dengan tujuan awal hanya untuk mencatat pengunjung pada *website* pribadi Rasmus Lerdorf. Pada rilis keduanya, ditambahkan *Form Interpreter*, sebuah *tools* untuk melakukan penerjemahan perintah SQL. Rilis kedua disebut dengan PHP/FI. Sejak itu, PHP mulai diteriman sebagai sebuah bahasa pemrograman baru yang sangat diminati. Terbukti pada pertengahan tahun 1997, tercatat sekitar 50.000 situs diseluruh dunia telah menggunakan PHP.

Dengan bertambah banyaknya pengguna PHP di seluruh dunia, maka PHP tidak memungkinkan lagi untuk dikelola oleh satu orang saja. Sehingga dibentuk sebuah tim pengembangan proyek *open source* ”*benevolent junta*”. Tim tersebut dipimpin oleh dua, Zeev Suraski dan Andi Gutmans. Keduanya lalu mendirikan sebuah perusahaan PHP dengan nama Zend (akronim dari Zeev Suraski dan Andi Gutmans). Selanjutnya Zend merilis versi PHP3 dan PHP4.

Tahun 1998 terjadi peningkatan penggunaan PHP yang sangat besar, bersamaan dengan naiknya popularitas penggunaan teknologi *open source*. Berdasarkan survei yang dilakukan Netcraft10, pada bulan Januari 2013 situs yang menggunakan PHP sudah mencapai 244 juta situs. *(Ahmad Solichin, 2015)*

* 1. **Laravel**

*Tulisan tentang laravel here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu framework, laravel secara umum, apa itu mvc, apa itu artisan, migrasi di laravel, arsitektur laravel yang terdiri dari controller, view, routes, midleware.*

* 1. **Github**

*Tulisan tentang github here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu revision control dan apa gunanya bagi sistem development dan developer*

* 1. **SSH**

*Tulisan tentang SSH here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu secure shell, dan cara mengakses secure shell untuk menginstall composer dan laravel serta menjalankan artisan.*

* 1. **Linux Server**

*Tulisan tentang Linux Server here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu linux server, dan apa itu apache dan nginx.*

* 1. **Apache**

*Tulisan tentang Apache here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu Apache, dan rule htaccess di apache*

* 1. **Rest-Api**

*Tulisan tentang rest-api here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu arsitektur rest-api, dan method apa saja yang ada di rest-api.*

* 1. **Postman**

*Tulisan tentang postman here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu testing tool untuk api dan cara kerja postman.*

* 1. **Composer**

*Tulisan tentang Composer here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu composer dan dependancy manager untuk php. Apa gunanya dan apa kelebihannya.*

* 1. **Node Package Manager (NPM)**

*Tulisan tentang NPM here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu composer dan dependancy manager untuk javascript. Apa gunanya dan apa kelebihannya.*

* 1. **Atom Text Editor**

Atom text editor here... Menyusul.... Hackable text editor!

* 1. **JavaScript**

*JavaScript* merupakan Bahasa pemrograman berbasis *Script*. *JavaScript* memiliki kemampuan untuk menciptakan kemampuan untuk menciptakan halaman *Web* yang dinamis serta didukung oleh banyak *Web Browser*. Hal ini menjadikan *JavaScript* sebagai Bahasa *Script* yang paling popular dan banyak digunakan oleh para programmer Web dalam pengembangan Web.

*JavaScript* berjalan di dalam kode HTML (*HyperText Markup Language*). Dengan menggunakan *JavaScript,* dapat membuat aplikasi yang interaktif pada halaman Web.

* 1. **jQuery**

jQuery adalah JavaScript library yang dirancang untuk meringkas kode-kode JavaScript, sehingga dapat menyederhanakan penulisan skrip program, sesuai dengan slogan “write less, do more”. jQuery pertama kali dirilis oleh John Resig pada tahun 2006, pada perkembangannya jQuery tidak hanya sebagai framework JavaScript, namun memiliki kelebihan antara lain:

1. Kemudahan mengakses dan memanipulasi elemen-elemen HTML.
2. Memanipulasi CSS.
3. Penanganan event HTML.
4. Efek-efek JavaScript dan animasi.
5. Memodifikasi elemen HTML DOM.

Sintak dasar jQuery $(selector).action(), tanda $ untuk mendefinisikan jQuery, jQuery selector digunakan untuk mendapatkan elemen HTML, action adalah tindakan yang dilakukan jQuery pada elemen () Contoh penggunaan jQuery untuk menyembunyikan elemen dengan id “test” sebagai berikut. $(“#test”).hide() Semua metode jQuery berada di dalam fungsi document.ready() yaitu perintah inisialisasi yang menunjukan dokumen telah siap ditampilkan dan sekaligus menjalankan perintah yang terdapat didalam fungsi.

* 1. **Ajax**

*Tulisan tentang Ajax here... Menyusul.... Garis besar saya akan jelaskan ttg apa itu ajax dan bagaimana ajax menangani method post. Dan apa itu asynchronous method.*

* 1. **Frontend**

*Tulisan tentang frontend here later....*

* 1. **React-Native**

Tulisan tentang pustaka untuk membangun aplikasi android nya here.... Later...

* 1. **Jenis-Jenis UML**

1. *Use Case Diagram*

*Use Case Diagram* adalah deskripsi dari sebuah sistem dari perspektif pengguna. *Use case diagram* bekerja dengan cara mendeskripsikan tipe interaksi antar pengguna. Sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (*Munawar, 2005*). *Use Case Diagram* terdiri dari:

1. *Actor*

*Actor* merupakan perwakilan dari orang diluar proses atau hal yang berinteraksi dengan sistem, subsistem ataupun *class.* Tiap *actor* berpartisipasi dengan satu atau lebih *use case*. *Actor* berpartisipasi dengan *use case* dengan pertukaran pesan.

1. *Use Case*

*Use case* merupakan lingkup sistem yang mengidentifikasi hal-hal yang seharusnya dilakukan oleh sistem. *Use case* berguna untuk menggambarkan suatu perilaku dari sistem tanpa mengungkapkan struktur internal dari sistem tersebut.

1. *Relationship*

*Relationship* berguna untuk menggambarkan hubungan antar *actor* dan *use case* dalam sistem. Jenis-jenis bentuk *relationship* seperti ditunjukkan pada Tabel 2.1.

Tabel 2.2 Tabel Bentuk Relationship

|  |  |  |
| --- | --- | --- |
| **Relationship** | **Fungsi** | **Notasi** |
| *Association* | Jalur komunikasi antar *aktor* dengan *use case* yang saling berpartisipasi. |  |
| *Extend* | Penambahan perilaku ke dalam *use case* dasar yang tidak tahu tentang hal tersebut. |  |
| *Usecase generalization* | Hubungan antara *use case* umum dengan *use case* yang lebih spesifik, yang mewarisi dan menambah fitur terhadapnya. |  |
| *Include* | Penambahan perilaku ke dalam use case dasar yang secara eksplisit menjelaskan penambahannya. | <<include>> |

1. *Class Diagram*

*Class diagram* adalah spesifikasi yang jika diinstalasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* memperlihatkan gambaran statik dari *class-class* dan hubungannya yang merupakan suatu *blue print* untuk objek (*Munawar,* 2005).

1. *Activity Diagram*

*Activity diagram* adalah teknik untuk menggambarkan logika procedural proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* dapat mendukung perilaku paralel sedangkan *flowchart* tidak (*Munawar, 2005*).

1. *Sequence Diagram*

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display,* dan sebagainya). Berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi (waktu) dan dimensi horizontal (objek-objek yang terkait).

1. *Collaboration Diagram*

*Collaboration Diagram* menjelaskan interaksi antar objek seperti *sequence diagram,* tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian pesan. Setiap pesan memiliki *sequence number,* di mana pesan dari level tertinggi memiliki nomor 1. Pesan dari tingkat yang sama memilik *prefix* yang sama.

1. *Component Diagram*

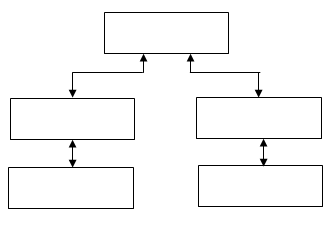
*Component Diagram* menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*dependency*). Komponen perangkat lunak adalah berisi kode, baik berisi kode sumber maupun kode biner, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time,* maupun *run time.* Umumnya komponen terbentuk dari beberapa *class* dan/atau *package,* tapi dapat juga dari komponen-komponen yang lebih kecil. (*Munawar, 2005*).

* 1. **Struktur Navigasi**

Struktur navigasi adalah alur yang digunakan dalam aplikasi yang dibuat. Sebelum menyusun aplikasi multimedia kedalam sebuah *software*, kita harus menentukan terlebih dahulu alur apa yang akan digunakan dalam aplikasi yang dibuat. Bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan aplikasi multimedia ada empat macam, yaitu Struktur Navigasi Hirarki, Liniear, Non Linear dan Campuran.

1. Struktur Navigasi Hirarki

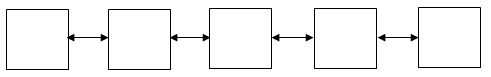
Struktur Navigasi Hirarki sering disebut struktur navigasi bercabang, yaitu merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data atau gambar pada *layer* dengan kriteria tertentu. Tampilan pada menu utama disebut *master page* (halaman utama satu), halaman tersebut mempunyai halaman percabangan yang disebut *slave page* (halaman pendukung) dan jika dipilih akan menjadi halaman kedua, begitu seterusnya, seperti pada Gambar 2.1.



Gambar 2.1 Struktur Navigasi Hirarki

1. Struktur Navigasi Linear

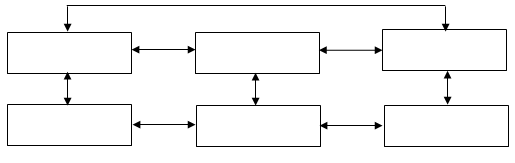
Struktur Navigasi Linear merupakan struktur yang mempunyai satu rangkaian cerita berurutan. Struktur ini menampilkan satu demi satu tampilan *layer* secara berurutan menurut aturannya, seperti pada Gambar 2.2.



Gambar 2.2 Struktur Navigasi Linear

1. Struktur Navigasi Non Linear

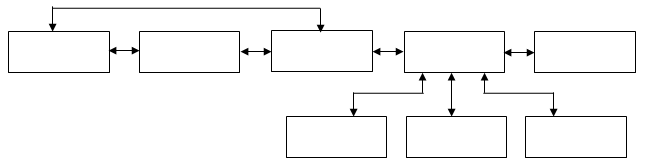
Struktur Navigasi Non Liniear (tidak terurut) merupakan pengembangan dari struktur Navigasi Linear, hanya saja pada struktur ini diperkenankan untuk membuat percabangan. Percabangan pada Struktur Non Linear berbeda dengan percabangan pada Struktur Hirarki, pada struktur ini kedudukan semua page sama, sehingga tidak dikenal adanya *master* atau *slave page*, seperti pada Gambar 2.3.



Gambar 2.3 Struktur Navigasi Non Linear

1. Struktur Navigasi Campuran

Struktur Navigasi Campuran (*Composite*) merupakan gabungan dari struktur sebelumnya dan disebut juga struktur Navigasi Bebas, maksudnya adalah jika suatu tampilan membutuhkan percabangan maka dibuat percabangan. Struktur ini paling banyak digunakan dalam pembuatan aplikasi multimedia, seperti pada Gambar 2.4.



Gambar 2.4 Struktur Navigasi Campuran