# lembar pengesahan

# abstrak

# abstract

# kata pengantar

# daftar isi

# daftar gambar

# daftar tabel

# daftar lampiran

# pendahuluan

Pada bab ini akan dijelaskan mengenai hal yang mendasari pembuatan website jurusan TI di SMK Negeri 4 Bandung ini, yaitu latar belakang masalah, tujuan pembuatan, rumusan dan batasan masalah, dan sistematika penulisan dalam pembuatan *website* jurusan TIsekolah ini.

## Latar Belakang

Internet merupakan hal yang biasa dikalangan pelajar, terutama pada abad ke-21 ini. Sebagian besar orang belakangan ini khususnya kalangan pelajar lebih banyak menghabiskan waktunya untuk membuka internet. Sebagian besar positif, yaitu mencari bahan pembelajaran, mencari informasi bermanfaat dari suatu situs web, bahkan membuka diri berhubungan sosial melalui dunia maya.

Internet tak lepas dari yang namanya website. Pengertian website/www/situs web sendiri merupakan kumpulan informasi pada beberapa *server* *computer* yang terhubung satu sama lain dalam jaringan internet. Beberapa instansi, perusahaan, dan kalangan formal lainnya menggunakan *website* untuk *company* *profile* mereka sudah bukan hal yang ganjil, karena melalui *website* tersebut mereka akan dipermudah dalam hal sosialisasi dan pengembangan perusahaan atau kalangan formal tersebut.

Salah satu contoh dari kalangan formal adalah dunia pendidikan. Pendidikan digolongkan ke dalam dua kategori, yaitu pendidikan formal dan informal. Pendidikan formal merupakan tempat mencari ilmu yang kurikulum, operasional, dan lama waktu belajarnya sudah ditetapkan oleh pemerintah dengan undang-undangnya. Salah satu dari sarana pendidikan formal tersebut adalah Sekolah Menengah Kejuruan Negeri 4 Bandung yang memiliki beberapa jurusan sesuai kurikulum yang berlaku.

Beberapa sekolah sudah memiliki *website* pribadi untuk menginformasikan tentang sekolah tersebut, salah satunya SMKN 4 Bandung. Website SMKN 4 Bandung yang ada memang sudah cukup informatif, baik secara akademik maupun nonakademik. Namun, info dari *website* tersebut hanya mencakup sebagian besar informasi secara umum tentang sekolah tersebut. *Website* yang ada berperan sebagai jembatan yang menghubungkan kalangan umum dengan kalangan sekolah secara umum. Setiap sekolah kejuruan memiliki bidang keahliannya masing-masing yang dijelaskan sekilas pada website SMK Negeri 4 Bandung. Berdasarkan hal tersebut, disusunlah suatu “Website Teknik Informatika SMK Negeri 4 Bandung Berbasis PHP dan MySQL” sebagai alat untuk menginformasikan secara terperinci secara akademik maupun nonakademik jurusan Teknik Informatika yang ada di SMK Negeri 4 Bandung, sebagai media publikasi, serta memperkenalkan jurusan TI di sekolah secara global agar mudah diakses oleh seluruh kalangan.

## Tujuan Pembuatan

Adapun tujuan pembuatan *website* ini adalah

1. Membuat Website TI SMK Negeri 4 Bandung dari beberapa sudut pandang pengguna.
2. Sebagai alat publikasi, promosi, dan sarana memperkenalkan jurusan TI di sekolah pada kalangan umum.
3. Mempermudah guru dan siswa dalam menyampaikan dan mendapatkan informasi yang harus segera disampaikan dalam waktu yang tidak memungkinkan tanpa harus bertemu muka secara langsung.

## Rumusan Masalah

Agar masalah yang dibahas diatas dapat dipahami dan lebih terarah, maka dibuatlah pertanyaan sebagai berikut:

1. Media seperti apakah yang tepat untuk mempublikasikan, untuk mempromosikan, dan memperkenalkan jurusan TI di sekolah?
2. Media seperti apakah yang mampu untuk membantu para siswa untuk memperoleh dan menyampaikan informasi mengenai jurusan TI di sekolahnya di luar waktu jam belajar?

## Batasan Masalah

Sesuai dengan judul tugas akhir ini, maka yang akan dibahas disini adalah tentang website TI SMK Negeri 4 Bandung yang dikhususkan pada perancangan sistem dengan batasan-batasan sebagai berikut:

1. *Website* ini dibangun menggunakan bahasa pemrograman berorientasi objek berbasis PHP dengan *database* berbasis MySQL.
2. *Website* ini diperuntukkan bagi warga SMK Negeri 4 Bandung dan umum, khususnya jurusan TI di SMK Negeri 4 Bandung.
3. Kalangan umum hanya bisa melihat fitur yang tersedia dan mengisi buku tamu.
4. Fitur yang terdapat dalam *website* ini meliputi manajemen:
   1. Berita dan Artikel,
   2. Galeri Gambar,
   3. Majalah Dinding,
   4. Agenda Sekolah,
   5. Pengguna,
   6. *Highlight*,
   7. *Polling*,
   8. Komentar,
   9. Buku Tamu, dan
   10. Statistik *website.*
5. Dibuat dengan menggunakan Netbeans IDE sebagai *text editor,* dan dijalankan menggunakan web browser.

## Sistematika Penulisan

|  |  |
| --- | --- |
| BAB I | PENDAHULUAN  Membahas latar belakang permasalahan, maksud dan tujuan, lingkup masalah, batasan masalah, identifikasi masalah, dan sistematika penulisan. |
| BAB II | TEORI DASAR  Menjelaskan tentang teori yang melandasi masalah dan referensinya. |
| BAB III | ANALISIS MASALAH  Berisi penjelasan tentang perancangan prosedur yang diusulkan (menjelaskan prosedur pelaksanaan pekerjaan yang baru dengan menyertakan komputer sebagai alat bantu proses pengolahan datanya). |
| BAB IV | PERANCANGAN APLIKASI  Berisi penjelasan tentang perancangan antar muka sistem yang akan diimplementasikan pada bab selanjutnya. |
| BAB IV | IMPLEMENTASI  Berisi implementasi struktur menu dan implementasi antar muka pengguna, admin, dan *homepage*. |
| BAB V | PENUTUP  Berisi kesimpulan dan saran pengembangan umum maupun pengembangan *website*/alat. |
| DAFTAR PUSTAKA | |
| LAMPIRAN-LAMPIRAN | |

# teori dasar

Pada bab ini akan dijelaskan mengenai materi-materi pendukung dalam pembuatan aplikasi ini, yaitu teori dasar tentang Teknik Informatika SMK Negeri 4 Bandung, teori tentang perancangan, teori tentang bahasa pemrograman, teori tentang *software,* dan sarana Pendukung pembuatan program.

## SMK Negeri 4 Bandung

Sesuai tuntutan pasar kerja, SMK Negeri 4 Bandung yang awalnya bernama STM Negeri 2 Bandung berdiri pada tahun 1962. Berawal dari jurusan Mesin, Bangunan, Listrik dan Kimia dengan lokasi di Jl. Ciliwung Bandung.

Pada tahun 1965 sekolah berpindah ke Jl. Kelenteng dan selanjutnya pada tahun 1969 sekolah berpindah ke Jl. Kliningan No. 6 Buahbatu Bandung sampai sekarang. Jurusan menjadi Listrik Instalasi, dan Elektronika Komunikasi.

Pada tahun pelajaran 1999/2000 STM Negeri 2 Bandung berubah nama menjadi SMK Negeri 4 Bandung. Program keahlian yang dibuka menjadi:

* Teknik Elektronika Komunikasi
* Teknik Listrik Pemakaian
* Teknik Instalasi Listrik
* Teknik Informatika dan Komunikasi

Pada tahun 2004 SMK Negeri 4 Bandung membuka Program Keahlian tersebut di atas menjadi:

* Teknik Audio Video
* Teknik Pemanfaatan Tenaga Liatrik
* Teknik Komputer dan Jaringan
* Rekayasa Perangkat Lunak

Pada tahun 2009 SMK Negeri 4 Bandung membuka program keahlian baru hingga memiliki 6 program keahlian, yaitu:

* Teknik Elektronika Audio Video
* Teknik Pemanfaaan Tenaga Listrik
* Teknik Otomasi Industri
* Teknik Komputer Jaringan
* Rekayasa Perangkat Lunak
* Multimedia

Visi dan Misi SMK Negeri 4 Bandung yang digunakan pada saat ini:

Visi: Menjadi Sekolah Menengah Kejuruan Unggulan di Jawa Barat yang berstandar Nasional dan Internasional.

Misi:

1. Memperoleh calon siswa melalui seleksi yang proporsional.
2. Mendidik, mengembangkan karakter dan bakat peserta didik melalui Program Mata Pelajaran Wajib,Kejuruan, Muatan Lokal serta Pengembangan diri.
3. Membangun Kepercayaan Masyarakat melalui keterbukaan manajemen dan keuangan sekolah.
4. Membangun kredibilitas dan akuntabilitas sekolah melalui administrasi yang tertib, bersih dan transparan.
5. Meningkatkan kepercayaan stakeholder melalui Program kejuruan dan kualitas kegiatan belajar mengajar.
6. Mengembangkan infrastruktur untuk mendukung proses kegiatan belajar mengajar dengan bantuan stakeholder.
7. Menghasilkan lulusan yang mampu bersaing dimasyarakat serta tidak melupakan budaya Jawa Barat.

Salah satu jurusan yang ada di SMK Negeri 4 Bandung ini adalah jurusan Teknik Informatika yang akan dibahas pada subbab ini.

### Jurusan Teknik Informatika

Jurusan Teknik Informatika merupakan jurusan yang terbilang baru di SMK Negeri 4 Bandung. Jurusan ini lahir pada tahun ajaran 1999/2000. Pada saat itu jurusan ini belum terpecah menjadi beberapa bidang seperti sekarang. Pada tahun ajaran 2004/2005 jurusan Teknik Informatika dibagi menjadi 2 bidang, yaitu Teknik Komputer Jaringan dan Rekayasa Perangkat Lunak. Pada tahun ajaran 2009/2010 sampai sekarang jurusan ini dibagi menjadi 3 jurusan seperti keterangan dibawah ini. Setiap jurusan memiliki bidang keahlian masing-masing dengan mata pelajaran kejuruan yang berbeda. Untuk jurusan Teknik Informatika di SMK Negeri 4 Bandung sampai angkatan masuk tahun 2011/2012 menggunakan program sekolah empat tahun, untuk tahun selanjutnya program sekolah menjadi pendidikan tiga tahun.

Teknik Informatika SMK Negeri 4 Bandung sendiri pada tahun ajaran 2009/2010 hingga sekarang dibagi menjadi tiga bidang berbeda, yaitu:

* Teknik Komputer Jaringan
* Rekayasa Perangkat Lunak
* Multimedia

## Internet

Internet berawal dari diciptakannya teknologi jaringan komputer. Jaringan komputer pada saat itu merupakan beberapa komputer yang terhubung satu sama lain dengan memakai kabel dalam satu lokasi, misalnya satu kantor atau gedung. Jaringan komputer itu berfungsi agar pengguna komputer bisa bertukar informasi dan data dengan pengguna komputer lainnya.

*ARPAnet (US Defense Advaced Research Project Agency)* atau Departemen Pertahanan Amerika pada tahun 1969 membuat jaringan komputer yang tersebar untuk menghindarkan terjadinya informasi terpusat, yang apabila terjadi perang dapat mudah dihancurkan. Jadi bila saru bagian dari sambungan jaringan terganggu dari serangan musuh, jalur yang melalui sambungan itu secara otomatis dipindahkan ke sambungan lainnya.

Setelah angkatan bersenjata Amerika, dunia pendidikan pun merasa sangat perlu mempelajari dan mengembangkan jaringan komputer. Salah satunya adalah *University of California at Los Angeles (UCLA)*. Akhirnya pada tahun 1970 internet banyak digunakan di universitas-universitas di Amerika dan berkembang pesat sampai saat ini. Agar pengguna komputer dengan merek dan tipe berlainan dapat saling berhubungan, maka para ahli membuat sebuah protokol (semacam bahasa) yang sama untuk dipakai di internet. Namanya *TCP (Trasmission Control Protocol)* dan *IP (Internet Protocol).*

## WWW (*World Wide Web*)

*World Wide Web (WWW)* lebih dikenal dengan web, merupakan salah satu layanan yang dapat dipakai oleh pemakai komputer yang terhubung ke internet.

Web pada awalnya adalah ruang informasi dalam internet. Dengan menggunakan teknologi hypertext, pemakai dituntun untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen web yang ditampilkan dalam browser web.

Saat ini internet identik dengan web, karena kepopuleran web sebagai standar interface pada layanan-layanan yang ada di Internet, dari awalnya sebagai penyedia informasi, kini digunakan juga untuk komunikasi dari email sampai dengan chatting, sampai dengan melakukan transaksi bisnis (*commerce*).

Saat ini web seakan lebih populer daripada email, walaupun secara statistik email masih merupakan aplikasi terbanyak yang digunakan oleh pengguna internet. Web lebih populer bagi khalayak umum dan pemula, terutama untuk tujuan pencarian informasi dan melakukan komunikasi email yang menggunakan web sebagai interfacenya.

Internet identik dengan web, karena popularitasnya sebagai penyedia informasi dan interface yang dibutuhkan oleh pengguna internet dari masalah informasi sampai dengan komunikasi. Informasi produk dari yang serius sampai dengan sampah, dari yang cuma-cuma sampai dengan yang komersial, semuanya ada..

Selain itu, web telah diadopsi oleh perusahaan sebagai bagian dari strategi teknologi informasinya, karena beberapa alasan berikut :

1. Akses informasi mudah.
2. Instalasi server lebih mudah.
3. Informasi mudah didistribusikan.
4. Bebas *platform* (Dapat disajikan diberbagai browser pada sistem operasi manapun).

### Lahirnya Web

Tahun 1989, Timothy Bernes-Lee seorang ahli komputer dari Inggris dan peneliri lain di *Europan Particle Physics Lab (Consei Europan pour la Recherche Nucleaire atau CERN)* di Geneva, Swiss, mengembangkan suatu cara untuk menyebarkan data antar koleganya menggunakan sesuatu yang disebut dengan hypertext. Pemakai di ERN dapat menampilkan dokumen pada layar komputer dengan mengunakan software browser baru.

Kode-kode khusus disisipkan ke dalam dokumen elektronik ini memungkinkan pemakai untuk meloncat dari satu dokumen ke dokumen lainnya pada layer dengan hanya memilih sebuah *hyperlink*.

Kemampuan internet dimasukkan kedalam browser ini sehingga lompat dari satu dokumen ke dokumen lain yang letaknya pada komputer remote. Seorang peneliti dapat juga mengirimkan sebuah file dari komputer remote ke sistem lokalnya, atau masuk ke dalam suatu sistem remote hanya dengan mengklik hyperlink, tidak perlu melalui mekanisme *FTP* atau *Telnet*. Jalan pintas *CERN* digunakan sebagai dasar dari yang disebut sekarang dengan *World Wide Web* dan berikut server browser webnya.

### Browser Web

Browser web adalah software yang digunakan untuk menampilakan informasi dari server web. Software ini kini telah dikembangkan dengan menggunakan *user* interface grafis, sehingga pemakai dapat dengan mudah melakukan poin dan klik untuk pindah antar dokumen.

Lynx adalah browser web yang masih menggunakan mode teks, yang akibatnya adalah tidak adanya gambar yang dapat ditampilkan. Lynx terdapat pada lingkungan *DOS (Disk Operating System)* dan *\*.nix (keluarga sistem operasi UNIX)*. Akan tetapi pekembangan dari browser mode teks ini tidaklah secepat browser web denga *GUI (Graphical User Interface)*.

Di saat perkembangan *WWW* ada dua browser web yang populer yaitu Internet Explorer (IE) dan Netscape Navigator. Namun saat ini bermunculan browser web lain yang turut meramaikan persaingan untku merebut para pengguna internet. Browser web yang telah banyak digunakan pada saat peluncurannya diantara lain adalah MSN, Opera dan Mozilla Firefox. Di antara browser web ini terdapat kelebihan dan kekurangan masing-masing sehingga pengguna dapat memilih sesuai keinginan dan kemudahan yang ditawarkan.

### Server Web

Server Web adalah komputer yang digunakan untuk menyimpan dokumen-dokumen web, komputer ini akan melayani permintaan dokumen web dari kliennya.

Browser web seperti IE atau Opera berkomunikasi melalui jaringan (termasuk jaringan internet) dengan server web, menggunakan HTTP. Browser akan mengirimkan request kepada server untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh server. Server memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protokol HTTP.

### Situs Web Statis dan Dinamis

Sekilas perbedaan antara situs web statis dengan situs web dinamis akan dijelaskan pada bagian ini. Pada pembuatan Website Teknik Informatika SMK Negeri 4 Bandung menggunakan Situs Web Dinamis.

1. Situs Web Statis

Situs web statis merupakan situs web yang memiliki isi tidak dimaksudkan untuk diperbarui secara berkala sehingga pengaturan ataupun pemutakhiran isi atas situs web tersebut dilakukan secara manual. Ada tiga jenis perangkat utilitas yang biasa digunakan dalam pengaturan situs web statis:

Editor teks merupakan perangkat utilitas yang digunakan untuk menyunting berkas halaman web, misalnya: Notepad atau TextEdit.

Editor WYSIWYG, merupakan perangkat lunak utilitas penyunting halaman web yang dilengkapi dengan antar muka grafis dalam perancangan serta pendisainannya, berkas halaman web umumnya tidak disunting secara lengsung oleh pengguna melainkan utilitas ini akan membuatnya secara otomatis berbasis dari laman kerja yang dibuat oleh pengguna. perangkat lunak ini misalnya: Microsoft Frontpage, Macromedia Dreamweaver.

Editor berbasis templat, beberapa utilitas tertentu seperti Rapidweaver dan iWeb, pengguna dapat dengan mudah membuat sebuah situs web tanpa harus mengetahui bahasa HTML, melainkan menyunting halaman web seperti halnya halaman biasa, pengguna dapat memilih templat yang akan digunakan oleh utilitas ini untuk menyunting berkas yang dibuat pengguna dan menjadikannya halam web secara otomatis.

1. Situs Web Dinamis

Situs web dinamis merupakan situs web yang secara spesifik didisain agar isi yang terdapat dalam situs tersebut dapat diperbarui secara berkala dengan mudah. Sesuai dengan namanya, isi yang terkadung dalam situs web ini umumnya akan berubah setelah melewati satu periode tertentu. Situs berita adalah salah satu contoh jenis situs yang umumnya mengimplementasikan situs web dinamis.

Tidak seperti halnya situs web statis, pengimplementasian situs web dinamis umumnya membutuhkan keberadaan infrastruktur yang lebih kompleks dibandingkan situs web statis. Hal ini disebabkan karena pada situs web dinamis halaman web umumnya baru akan dibuat saat ada pengguna yang mengaksesnya, berbeda dengan situs web statis yang umumnya telah membentuk sejumlah halaman web saat diunggah di server web sehingga saat pengguna mengaksesnya server web hanya tinggal memberikan halaman tersebut tanpa perlu membuatnya terlebih dulu.

Untuk memungkinkan server web menciptakan halaman web pada saat pengguna mengaksesnya, umumnya pada server web dilengkapi dengan mesin penerjemah bahasa skrip (PHP, ASP, ColdFusion, atau lainnya), serta perangkat lunak sistem manajemen basisdata relasional seperti MySQL.

Struktur berkas sebuah situs web dinamis umumnya berbeda dengan situs web statis, berkas-berkas pada situs web statis umumnya merupakan sekumpulan berkas yang membentuk sebuah situs web. Berbeda halnya dengan situs web dinamis, berkas-berkas pada situs web dinamis umumnya merupakan sekumpulan berkas yang membentuk perangkat lunak aplikasi web yang akan dijalankan oleh mesin penerjemah server web, berfungsi memanajemen pembuatan halaman web saat halaman tersebut diminta oleh pengguna.

## Analisis dan Perancangan

Pada bagian analisis dan perancangan ini akan dijelaskan mengenai teori analisis dan perancangan sistem menggunakan *Unified Modeling Language (*UML*)* dan *Flowchart* serta perancangan basis data*.*

### Analisis dan Perancangan Sistem

Dalam suatu pembangunan sistem, sebelumnya harus dilakukan analisis dan perancangan sistem terlebih dahulu. Analisis dan perancangan sistem sangatlah penting dalam suatu pembuatan sistem. karena:

1. Dapat mengetahui seberapa besar sistem yang akan dibuat.
2. Estimasi waktu pengerjaan yang lebih akurat.
3. Meminimalisasi biaya.

Analisis adalah proses dimana sistem yang sedang belajar dipelajari dan sistem pengganti diusulkan. Tujuan utama dari analisis ini adalah untuk memahami dan mendokumentasikan kebutuhan bisnis. Ada lima aktifitas utama dalam fase ini, yaitu:

1. Pengumpulan Informasi.
2. Mendefinisikan Kebutuhan Sistem.
3. Memprioritaskan Kebutuhan.
4. Menyusun dan Mengevaluasi Alternatif.
5. Mengulas Kebutuhan dengan Pihak Manajemen.

Perancangan adalah proses menulis, menguji dan memperbaiki (*debug*), dan memelihara kode yang membangun sebuah program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau pekerjaan sesuai dengan keinginan si pemrogram. Untuk dapat melakukan pemrograman, diperlukan keterampilan dalam algoritma, logika, bahasa pemrograman dan di banyak kasus, pengetahuan-pengetahuan lain seperti penyampaian informasi yang padat dan berbobot.

### *Unified Modelling Language* (UML)

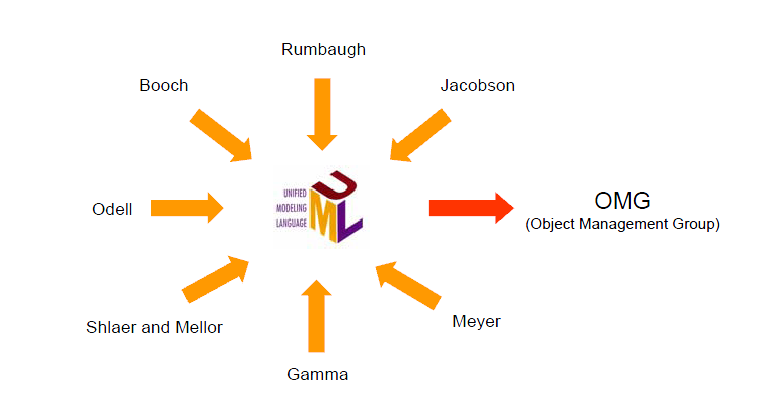
*Unified Modelling Language*(UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan peranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram peranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object* *Modeling* *Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.



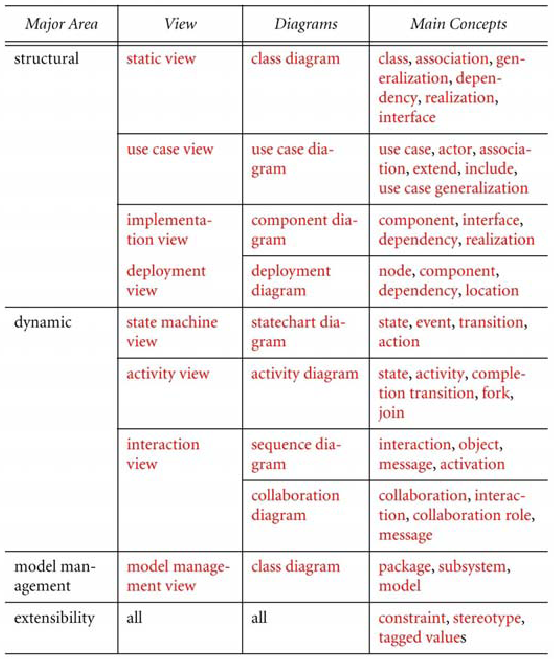
Gambar 2.1 Pengkoordinasian oleh *Object Management Group* (OMG)

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group*(OMG – <http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

UML sendiri telah menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek untuk membantu pemodelan suatu aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual* *Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence* *Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*.
4. *Collaboration* *Diagram* untuk memodelkan interaksi antar *objects*.
5. *State* *Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity* *Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* di dalam sistem.
7. *Class* *Diagram* untuk memodelkan struktur kelas.
8. *Object* *Diagram* untuk memodelkan struktur *object*.
9. *Component* *Diagram* untuk memodelkan komponen *object*.
10. *Deployment* *Diagram* untuk memodelkan distribusi aplikasi.

Tabel 2.1 Konsep Dasar UML



Yang akan penulis gunakan dalam memodelkan pembangunan *Website* adalah *Use Case Diagram, Activity Diagram,* dan *Class Diagram.*

### Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan dalam perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Basis data merupakan salah satu komponen utama dalam sistem informasi, karena merupakan basis dalam penyediaan informasi bagi para pemakai (Fathansyah, 1999; Post, 1999).

Jika dibayangkan, basis data mirip dengan lemari di ruang administrasi sekolah yang menyimpan berbagai arsip. Masing-masing jenis arsip dikelompokkan, diatur dan disimpan pada tempat yang telah ditentukan. Sehingga akan ada kelompok arsip siswa, arsip guru, arsip mata pelajaran, arsip keuangan, dan lain-lain. Perbedaannya hanya pada media penyimpanannya. Kalau lemari arsip menggunakan lemari dari kayu, besi atau plastik, sedangkan basis data menggunakan media penyimpan elektronik seperti disk (hard disc, CD, atau tape).

Satu hal penting yang harus diperhatikan, basis data bukan hanya sekedar penyimpanan data secara elektronik. Tidak semua penyimpanan data elektronik bisa disebut basis data. Apabila penyimpanan itu tidak menggunakan prinsip pengaturan, pemisahan atau pengorganisasian maka kita tidak dapat menyebut penyimpanan data tersebut sebagai basis data.

Prinsip utama dalam basis data adalah konsep independensi data yaitu pemisahan data dari program aplikasinya (Lewis et al., 2002; Post, 1999). Sedangkan tujuan utama dalam basis data adalah membantu pengguna dalam abstraksi suatu sistem. Ada tiga level abstraksi yang biasanya digunakan yaitu *physical* *level*, *conceptual level* dan *view level*. *Physical level* menunjukkan bagaimana data akan disimpan. *Conceptual* *level* berkaitan dengan data apa yang akan disimpan dan bagaimana hubungan antar data tersebut. *View* *level* merupakan level tertinggi yang menjelaskan bagian-bagian basis data pada pengguna tertentu (Ramakrishnan and Gehrke, 2000). Basis data mempunyai beberapa kriteria penting, yaitu

1. Bersifat data *oriented* dan bukan program *oriented*.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara-cara yang berbeda.

Secara bertingkat, operasi dasar basis data meliputi:

1. Pembuatan basis data baru (*create database*). Operasi ini sama dengan pembuatan atau pembelian lemari arsip yang baru.
2. Penghapusan basis data (*drop database*). Operasi ini sama dengan pengrusakan atau penghancuran lemari arsip.
3. Pembuatan tabel baru (*create table*). Operasi ini sama dengan penambahan kelompok arsip baru. Operasi ini baru bisa dijalankan jika basis data telah dibuat.
4. Penghapusan tabel (*drop table*). Operasi ini sama dengan pengrusakan kelompok arsip lama. Operasi ini baru bisa dijalankan jika tabel telah ada pada suatu basis data.

## Bahasa Pemrograman

Bahasa pemrograman, atau sering di istilah kan juga dengan [bahasa komputer](http://id.wikipedia.org/wiki/Bahasa_komputer), adalah teknik komando/instruksi standar untuk memerintah [komputer](http://id.wikipedia.org/wiki/Komputer). Bahasa pemrograman ini merupakan suatu himpunan dari aturan [sintaks](http://id.wikipedia.org/wiki/Sintaks) dan [semantik](http://id.wikipedia.org/wiki/Semantik) yang dipakai untuk mendefinisikan [program komputer](http://id.wikipedia.org/wiki/Program_komputer). Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis[langkah](http://id.wikipedia.org/wiki/Algoritma) apa secara persis yang akan diambil dalam berbagai situasi.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

1. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110
2. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (Inggris: [*Assembly*](http://id.wikipedia.org/wiki/Assembly)), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode *mnemonic*), contohnya MOV, SUB, CMP, JMP, JGE, JL, LOOP, dsb.
3. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&, ||, dsb.
4. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya begin, end, if, for, while, and, or, dsb.

### HTML (*HyperText* *Markup* *Language*)

Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal dengan sebagai web page. Dokumen HTML merupakan dokumen yang disajikan dalam browser web surfer. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet.

Ada dua cara untuk membuat sebuah web page diantaranya adalah dengan web editor (misalnya Macromedia Dreamweaver atau Microsoft Frontpage) atau dengan editor teks biasa (misalnya Notepad atau Editplus).

1. Penamaan Dokumen HTML

Penamaan dokumen HTML dapat dilakukan dengan memilih suatu nama, sembarang nama, kemudian menambahkan sebuah ekstensi “.htm” atau “.html” (tanpa kutip tidak perlu dituliskan karena digunakan sebagai penjelas saja).

Ekstensi dokumen HTML yang menggunakan 3 karakter awalnya adalah untuk mengakomodasikan sistem penamaannya yang ada dalam sistem operasi DOS. Nama dokumen pada beberapa sistem operasi bersifat *case sensitive* (membedakan huruf besar dan huruf kecil). Nama dokumen yang sama tetapi dituliskan dengan case yang berbeda akan dianggap sebagai dokumen yang berbeda, misalnya dokumen html akan berbeda dengan DOKUMEN.html. Kasus *case* *sensitive* akan dijumpai pada dokumen web yang di *hosting* di dalam *server* yang berbasis \*nix (keluarga sistem operasi UNIX).

1. Elemen dan Tag HTML

Dokumen HTML disusun oleh elemen-elemen. “Elemen” merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Untuk menandai berbagai elemen dalam suatu dokumen HTML, digunakan tag. Tag HTML terdiri atas sebuah kurung sudut kiri (<, tanda lebih kecil), sebuah nama tag, dan sebuah kurung sudut kanan (>, tanda lebih besar). Tag umumnya berpasangan (misalnya <H1> dengan </H1>), tag yang menjadi pasangan selalu diawali dengan karakter garis miring. Tag yang pertama menunjukkan tag awal yang berartu awak elemen, dan yang kedua menunjukan tag akhir, berarti akhir elemen.

Nama elemen ditunjukkan dengan nama tag nya. Suatu elemen didalam dokumen HTML harus ditandai dengan penulisan tag nya berpasangan. Ada beberapa elemen yang tidak mengharuskan tag nya dituliskan secara berpasangan, elemen tersebut diantaranya adalah:

* Paragraf dengan tag <p>
* Ganti baris – line break dengan tag <br>
* Garis datar – horizontal rule dengan tag <hr>
* List item dengan tag <li>

Secara umum suatu elemen dalam dokumen HTML yang dinyatakan dengan tagnya, dituliskan : <nama tag>\_</nama tag>. Penulisan penamaan tag bebas, dapat menggunakan huruf besar, huruf kecil, ataupun campuran (tidak *case sensitive*). Tetapi untuk antisipasi pada standar penulisan tag, direkomendasikan untuk menuliskan tag dengan menggunakan huruf kecil semuanya.

Tag *mark up* ini digunakan sebenarnya memberi tahu browser web bagaimana memperlakukan atau menampilkan (memformat) halaman dokumen tersebut dalam browser.

Dalam membuat suatu dokumen HTML dibutuhkan elemen yang dinyatakan dengan tag <html>, <head> dan <body> berikut tag-tag pasangannya. Setiap dokumen terdiri atas tag head dan body. Elemen head berisi teks yang sebenarnya tersusun dari link, grafik, paragraf dan elemen lainnya.

|  |
| --- |
| **Contoh bahasa pemrograman menggunakan html (belajar.html)** |
|  |

### PHP (PHP *Hypertext* *Preprocessor*)

PHP adalah kependekan dari “PHP *Hypertext Preprocessor*” merupakan sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java, dan Perl ditambah beberapa fungsi spesifik dari PHP itu sendiri. Tujuan utama untuk penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dengan cepat.

1. Hubungan PHP dengan HTML

Halaman web biasanya disusun dari kode-kode html yang disimpan dalam sebuah file berekstensi .html. File html ini dikirimkan oleh server (atau file) ke browser, kemudian browser menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program php, program ini harus diterjemahkan oleh web-server sehingga menghasilkan kode html yang dikirim ke browser agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode html sehingga dapat langsung ditampilkan bersama dengan kode-kode html tersebut. Program php dapat ditambahkan dengan mengapit program tersebut di antara tanda . Tanda-tanda tersebut biasanya disebut tanda untuk escaping (kabur) dari kode html. File html yang telah dibubuhi program php harus diganti ekstensi-nya menjadi .php3 atau .php. PHP merupakan bahasa pemograman web yang bersifat server-side, di mana script-nya menyatu dengan HTML dan berada server. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di server tetapi disertakan HTML biasa. PHP dikenal sebgai bahasa scripting yang menyatu dengan tag HTML, dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis seperti *ASP (Active Server Pages)* dan *JSP (Java Server Pages)*.

PHP pertama kali dibuat oleh Rasmus Lerdroft, seorang programmer C. Semula PHP digunakannya untuk menghitung jumlah pengunjung di dalam webnya.Kemudian ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan script PERL yang dibuatnya untuk membuat halaman webnya menjadi dinamis. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah dapat mengakses *database* dan dapat terintegrasi dengan HTML.

Pada tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang softwarenya.Versi terbaru, yaitu PHP 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya Zend Engine yang dibuat oleh Zend Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP scripting engine. Yang lainnya adalah build in HTTP session, tidak lagi menggunakan library tambahan seperti pada PHP. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan web server.

1. Kelebihan PHP

Ketika e-commerce semakin berkembang, situs-situs yang statis pun semakin ditinggalkan, karena dianggap sudah tidak memenuhi keinginan pasar, padahal situs tersebut harus tetap dinamis. Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar designer web banyak beralih ke bahasa *server-side* scripting yang lebih dinamis seperti PHP. Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan system *database* di dalam web. Sistem *database* yang dapat didukung oleh PHP antara lain:

* Oracle,
* MySQL,
* Sysbase,
* PostgreSQL.

PHP dapat berjalan di berbagai system operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh.

Untuk menjalankan sistem PHP, dibutuhkan tiga komponen. Yang pertama adalah webserver, karena PHP adalah sebuah bahasa pemrograman *server-side* (dijalankan di sisi server). Yang kedua adalah program PHP itu sendiri. Program inilah yang akan memproses skrip-skrip PHP kita sehingga dapat berjalan dan mengeluarkan hasil. Program PHP dapat berupa program yang berdiri sendiri maupun berupa modul pada webserver. Yang ketiga adalah server *database*. Ini sebetulnya tidak wajib, tapi amat amat umum dipakai bersama PHP, untuk menyimpan data-data kita/pengunjung agar nanti dapat diproses kembali.

|  |
| --- |
| **Contoh bahasa pemrograman menggunakan php (belajar.php)** |
|  |

### CSS (*Cascading* *Style* *Sheet*)

*Cascading Style Sheets* (CSS) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh *World Wide Web Consortium* (W3C). CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi.

CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca braille. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

CSS pertama kali dikembangkan pada tahun 1996, dimana *World Wide Web Consortium* atauW3C sebuah lembaga konsorsium untuk standarisasi web menyusun draft proposal untuk membuat CSS. Setelah proposal diterima, berjalanlah proyek pembuatan CSS ini hingga muncul CSS level 1. CSS level 1 ini mendukung beberapa pengaturan seperti:

* Jenis *font* dan ketebalan *font*,
* Warna, *background* dan teks,
* Posisi teks dan gambar,
* *Margin*, *border* dan *padding*.

Dipertengahan tahun 1998, W3C kembali mengembangkan CSS dari level 1 menjadi CSS level 2. CSS1 dikembangkan untuk pemformatan dokumen HTML, sedangkan CSS2 pengembangannya berpusat kepada dokumen media agar mampu tampil dalam web. CSS3 yang sekarang sudah bukan hal aneh, CSS3 sudah banyak memenuhi tampilan web karena fitur yang lebih *user friendly,* interaktif, dan efisien terhadap pengguna. CSS3 juga bisa mengurangi ukuran file yang di-load dan lebih ringan, secara otomatis mengurangi bandwith inbound/outbound situs.

|  |
| --- |
| **penggunaan css dalam halaman html (inlinestyle.html)** |
|  |

## Sarana Pendukung

Subbab ini menjelaskan tentang beberapa aplikasi maupun kerangka kerja yang membantu proses pembuatan *website* ini.

### Zend Framework

Pada pembangunan aplikasi web ini menggunakan kerangka kerja yang disebut Zend Framework. Pada subbab ini akan penulis jelaskan apa itu *framework* pada umumnya dan akan mengerucut kepada penggunaan Zend Framework.

1. Framework

*Framework* dapat diartikan sebagai alat yang digunakan untuk membantu pekerjaan. Karena untuk pembuatan website maka *framework* disini dapat diartikan sebagai alat yang dapat digunakan untuk mempermudah pembuatan website. Jika dengan CMS maka kita tinggal menjalankan saja tidak perlu lagi memikirkan untuk menulis kode program sendiri, tetapi tidak demikian dengan *framework*. Menggunakan *framework* kita masih harus menulis kode, bedanya kode-kode yang kita tulis harus menyesuaikan dengan lingkungan *framework* yang kita gunakan. Memang konsekuensinya kita harus belajar lagi lingkungan pengembangan berdasarkan *framework* yang kita gunakan, tetapi hal itu akan terbayar setelah kita menguasai dan bisa menggunakan *framework* tersebut.

Sebuah *framework* selain menyediakan lingkungan pengembangan sendiri-sendiri, juga menyediakan berbagai macam fungsi siap pakai yang bisa kita gunakan dalam pembuatan website. Sehingga tidak perlu kaget jika akan banyak kode atau fungsi yang terlihat tidak seperti biasanya, karena fungsi-fungsi tersebut merupakan fungsi bawaan *framework* dan bukan fungsi asli dari PHP. Fungsi tersebut terkadang merupakan pengembangan atau penyesuaian fungsi asli PHP agar lebih mudah digunakan atau agar lebih sesuai dengan kebutuhan pengguna.

1. Pengertian Zend Framework

ZENDFRAMEWORK merupakan PHP *Framework* berbasisMVC. Mengutip dari situs [Zend Framework](http://framework.zend.com/).

*“Zend**Framework is based on simplicity, object-oriented best practices, corporate friendly licensing, and a rigorously tested agile codebase. Zend Framework is focused on building more secure, reliable, and modern Web 2.0 applications & web services, and consuming widely available APIs from leading vendors like Google, Amazon, Yahoo!, Flickr, as well as API providers and cataloguers like StrikeIron and ProgrammableWeb.”*

“ZendFramework didasarkan pada kesederhanaan, berorientasi objek terbaik, perusahaan berlisensi, dan basis kode diuji ketat. Zend Framework difokuskan untuk membangun web agar lebih aman, handal, dan modern *Web 2.0 applications* dan layanan web,  dan memakai banyak API yang banyak tersedia dari vendor terkemuka seperti Google, Amazon, Yahoo, Flickr, serta penyedia API dan *cataloguers* seperti *StrikeIron* dan *Programmable Web*.”

Zend Framework adalah *open source* kerangka kerja yang berorientasi objek fitur komponen yang cukup sederhana dan nyaman untuk digunakan. Ini membantu PHP pengembang untuk dengan mudah membuat kompleks aplikasi web dengan menyediakan *Model-View-Controller* (MVC) desain yang dengan demikian membuat kode dapat digunakan kembali. Zend Framework merupakan Kerangka Keuntungan membantu dalam menciptakan Aplikasi web PHP dengan mudah dan meminimalisasi kerumitan. Ada sejumlah keuntungan yang melekat jika Zend Framework digunakan secara konstruktif dan dengan cara yang tepat. Salah satu keuntungan yang paling menonjol adalah Zend Framework yang dikenal karena kesederhanaan, menyediakan pengembang PHP dengan sebuah perpustakaan komponen ringan yang menawarkan sebanyak 80% fungsi dan bahkan memungkinkan seseorang untuk menyesuaikan 20% sisanya.

1. Fitur-fitur Zend Framework

Syarat menggunakan Zend Framework antara lain:

Kompatibel dengan PHP 5.1.4 (versi diatasnya).

Sistem berbasis *Model View Controller.*

Membutuhkan sebuah *web server* yang mendukung fungsi mod\_rewrite, co: menggunakan apache 2.

Beberapa Keunggulan dari Zend Framework antara lain:

1. Lisensi *Open Source* berbasis BSD (*Berkeley Software Distribution*) yang fleksibel baik untuk pengembangan aplikasi open source maupun komersial.
2. Dokumentasi lengkap (lebih dari 500 halaman).
3. Proses pengembangan Zend Framework menyertakan unit test yang lengkap, mencakup 84% kode yang ada.
4. Library yang menyediakan fungsi yang lengkap sehingga hampir semua kebutuhan pengembangan aplikasi web dapat terpenuhi.
5. Mendukung pengembangan aplikasi MVC.
6. URL yang bagus dan *search engine friendly*.
7. Peningkatan ketepatan, mudah dibaca dan lebih lengkap contohnya.
8. Penambahan cakupan terhadap inti-inti framework.
9. Pengembangan web aplikasi tutorial.

Macam-macam library yang disediakan oleh Zend Framework:

* Zend\_Acl
* Zend\_Authentication
* Zend\_Session
* Zend\_Cache, Zend\_Config, Zend\_Console\_Getopt, Zend\_Log, Zend\_Memory
* Zend\_Debug, Zend\_Environment, Zend\_Loader, Zend\_Registry, Zend\_Version
* Zend\_Filter, Zend\_Validate
* Zend\_Date
* Zend\_Locale
* Zend\_Measure
* Zend\_Translate
* Zend\_Json, Zend\_Pdf
* Zend\_Mail, Zend\_Mime
* Zend\_Search\_Lucene
* Zend\_Controller, Zend\_Controller\_Action, Zend\_Controller\_Dispatcher, Zend\_Controller\_Plugin, Zend\_Controller\_RewriteRouter, Zend\_View
* Zend\_Http\_Request, Zend\_Http\_Response
* Consuming services: Zend\_Feed, Zend\_Rest\_Client, Zend\_Service, Zend\_XmlRpc\_Client, Zend\_Gdata, Zend\_Http\_Client
* Exposing services: Zend\_Http\_Server, Zend\_Rest\_Server, Zend\_Server\_Documentor, Zend\_Server\_Reflection, Zend\_Soap\_Server, Zend\_XmlRpc\_Server
* Zend\_Uri

1. Metode MVC (Model View Controller)

Seperti sudah disebutkan di muka bahwa Zend Framework menerapkan lingkungan pengembangan dengan metode MVC (*Model View Controller*). MVC memisahkan antara logika pembuatan kode dengan pembuatan *template* atau tampilan *website*. Penggunaan MVC membuat pembuatan sebuah proyek *website* menjadi lebih terstruktur dan lebih sederhana.

Secara sederhana konsep MVC terdiri dari tiga bagian yaitu bagian *Model*, bagian *View* dan bagian *Controller*. Didalam website dinamis setidaknya terdiri dari 3 hal yang paling pokok, yaitu basis data, logika aplikasi dan cara menampilkan halaman *website*. Tiga hal tersebut direpresentasikan dengan MVC yaitu *model* untuk basis data, *view* untuk cara menampilkan halaman *website* dan *controller* untuk logika aplikasi.

1. Model

Merepresentasikan struktur data dari *website* yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks atau file xml. Biasanya didalam model akan berisi *class* dan fungsi untuk mengambil, melakukan *update* dan menghapus data *website*. Karena sebuah *website* biasanya menggunakan basis data dalam menyimpan data maka bagian *Model* biasanya akan berhubungan dengan perintah-perintah *query* SQL. *Model* bisa dibilang khusus digunakan untuk melakukan koneksi ke basis data oleh karena itu logika-logika pemrograman yang berada didalam model juga harus yang berhubungan dengan basis data. Misalnya saja pemilihan kondisi tetapi untuk memilih melakukan query yang mana.

1. View

Merupakan informasi yang ditampilkan kepada pengunjung *website*. Sebisa mungkin di dalam *View* tidak berisi logika-logika kode tetapi hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. *View* bisa dibilang adalah halaman *website* yang dibuat menggunakan HTML dengan bantuan CSS atau JavaScript. Didalam *view* jangan pernah ada kode untuk melakukan koneksi ke basis data. *View* hanya dikhususkan untuk menampilkan data-data hasil dari *model* dan *controller*.

1. Controller

*Controller* merupakan penghubung antara *Model* dan *View*. Didalam *Controller* inilah terdapat *class* dan fungsi-fungsi yang memproses permintaan dari *View* kedalam struktur data didalam *Model*. *Controller* juga tidak boleh berisi kode untuk mengakses basis data. Tugas *controller* adalah menyediakan berbagai variabel yang akan ditampilkan di *view*, memanggil *model* untuk melakukan akses ke basis data, menyediakan penanganan *error*, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap *input*.

Jadi secara singkat urutan dari sebuah *request* adalah sebagai berikut : *user* berhubungan dengan *view*, dimana didalam *view* inilah semua informasi ditampilkan. Saat *user* melakukan permintaan atau *request*, misal klik tombol maka *request* tersebut akan diproses oleh *Controller*. Apa yang harus dilakukan, data apa yang diinginkan, apakah ingin melihat data, atau memasukan data atau mungkin melakukan validasi data terlebih dahulu, semua diproses oleh *Controller*. Kemudian *Controller* akan meminta *Model* untuk menyelesaikan *request*, entah itu melakukan *query* atau apapun. Dari *Model*, data akan dikirim kembali untuk di proses lebih lanjut di dalam *Controller* dan baru dari *Controller* data akan ditampilkan di *View*.

### Netbeans IDE

**NetBeans** mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java dan sebuah *Integrated Development Environment*(IDE) yang dibangun menggunakan platform NetBeans.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut ‘modul’. Sebuah modul adalah suatu arsip Java (Java *archive*) yang memuat kelas-kelas Java untuk berinetraksi dengan *NetBeans Open API* dan file manifestasi yang mengidentifikasinya sebagai modul.

Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan *powerful*.

### XAMPP

XAMPP adalah perangkat lunak (APLIKASI) bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (*General Public License*) dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

# Analisis Masalah

Pada bab ini akan dijelaskan mengenai analisis dari permasalahan-permasalahan yang telah disampaikan pada bab pertama. Terbagi menjadi 2 bagian, yaitu, analisis sistem yang sedang berjalan, perancangan proses dan procedural sistem yang akan dibuat.

## Analisis Sistem yang Sedang Berjalan

Analisis sistem yang sedang berjalan merupakan bagian dari suatu proses untuk mengajukan ataupun membuat suatu sistem. Dalam subbab ini akan digambarkan analisis sistem yang sedang berjalan menggunakan pemodelan *use case diagram* dari sistem informasi jurusan TI SMK Negeri 4 Bandung yang masih berjalan secara konvensional.



Gambar 3.1 *Use Case* Sistem yang sedang berjalan

## Perancangan Proses dan Prosedural

Dari pembuatan Website TI SMK Negeri 4 Bandung akan dikembangkan sistem banyak pengguna (*multi-user system*) dimana pengguna yaitu, admin, guru, siswa dan pengguna diluar warga SMK Negeri 4 Bandung dengan kelayakan mengakses fitur yang dibatasi sesuai *user.*  perancangan proses website ini menggunakan *Use Case Diagram* dan dikelompokan sesuai user menggunakan *Activity Diagram*. Perancangan pembuatan program aplikasi web ini juga akan dimodelkan menggunakan *Class Diagram*.

### *Use Case Back-End* (*User*: Admin(Guru) dan Siswa)

Pada halaman *backend* hanya *user* yang mempunyai hak akses admin, guru, dan siswa yang dapat mengakses halaman admin sesuai dengan *role*nya masing-masing. *user* akan dipersilahkan untuk *login* saat akan memasuki halaman admin.



Gambar 3.2 Permodelan *Use Case Back-End* (Halaman Admin)

### *Use Case Front-End* (Admin(Guru), Siswa, dan Pengunjung)

Pada halaman *frontend* semua *user* yang mempunyai hak akses admin, guru, siswa ataupun pengunjung tanpa *login* dapat mengakses halaman ini sesuai dengan *role*nya masing-masing. *user* akan dipersilahkan untuk login saat ingin menambahkan komentar ataupun mengakses laman admin.



Gambar 3.3 Permodelan *Use Case Front-End* (Halaman Depan)

### *Activity Diagram* Siswa

*Activity Diagram* Siswa adalah kegiatan apa saja yang bisa siswa lakukan pada Website TI SMK Negeri 4 Bandung. Siswa yang tidak mempunyai akun hanya bisa mengakses halaman depan saja, sedangkan siswa yang sudah memiliki akun dapat mengakses bagian admin yang di khususkan untuk siswa.



Gambar 3.4 *Activity Diagram* Siswa

### *Activity Diagram* Guru

### *Activity Diagram* Admin

### *Activity Diagram* Pengunjung

# perancangan Aplikasi

Pada bab ini akan dijelaskan bagaimana analisis masalah dan perancangan prosedur yang diusulkan dari rencana pembuatan Website TI SMK Negeri 4 Bandung akan disambungkan kedalam perancangan basis data, perancangan antar muka sistem, kebutuhan sistem serta tabel jadwal pembuatan Website TI SMK Negeri 4 Bandung.

## Perancangan Peta Situs (*Site Map*)

Perancangan *site map* atau peta situs merupakan penggambaran umum secara keseluruhan dari suatu aplikasi pembuatan web maupun *desktop*. Perancangan ini berasal dari analisis masalah yang diajukan untuk menggantikan sistem lama yang sudah ada dan telah dibuat pemodelannya secara diagram pada bab sebelumnya. Dibawah ini akan digambarkan sebuah Struktur Menu dari Website TI SMK Negeri 4 Bandung.



Gambar 4.1 Peta Situs Website SMK Negeri 4 Bandung

## Perancangan Antar Muka Sistem

Perancangan antar muka sistem adalah peracangan yang disusun untuk membantu merealisasikan suatu pekerjaan agar lebih mudah. Subbab ini akan dibagi lagi menjadi beberapa bagian perancangan yang sudah dikelompokkan sesuai fungsi agar lebih mudah dipahami. Dibawah ini akan disertakan perancangan-perancangan tersebut dalam bentuk visual agar lebih mudah dimengerti.

### Perancangan Halaman

Perancangan halaman adalah bagian perencanaan untuk menyederhanakan proses pembuatan suatu aplikasi website. Dalam hal ini perancangan halaman biasanya dibuat dalam dua sudut pandang yang berbeda, yaitu halaman admin atau biasa disebut *back-end* dan halaman depan biasa disebut *homepage* atau *front-end*. Dibawah ini akan dirancang dua halaman tersebut dalam fokus fungsi berbeda.

A. Halaman Admin (*Back-End*)

Halaman admin atau biasa disebut *back-end* adalah halaman kendali penuh atas apa yang bisa dilakukan *user* tertentu dalam suatu situs web. Website TI SMK Negeri 4 Bandung ini membagi akses untuk bisa masuk ke halaman ini menjadi tiga bagian, yaitu:

* *User* dengan *role* Siswa, kendali pada user ini dibatasi hanya dapat menyampaikan informasi melalui artikel yang harus disetujui oleh *role* yang lebih berhak (dalam hal ini adalah guru dan admin).
* *User* dengan *role* Guru, keterbatasan kendali pada user ini adalah dapat menyampaikan informasi melalui artikel, berita, dan galeri tanpa harus disetujui oleh *role* yang lebih berhak. Guru juga dapat menyetujui suatu artikel yang dibuat oleh siswa sehingga artikel tersebut pantas untuk ditampilkan pada *homepage* Website TI SMK Negeri 4 Bandung.
* *User* dengan *role* Admin, merupakan satu-satunya *user* yang dapat melakukan segala bentuk penambahan informasi, menyortir informasi yang berhak dan tidak berhak tampil, dan kendali penuh atas seluruh akun yang ada di Website TI SMK Negeri 4 Bandung.

Rancangan Halaman ini yang nantinya akan diimplementasikan pada pembuatan Website TI SMK Negeri 4 Bandung sesudah melakukan login terlebih dahulu. Dibawah ini merupakan rancangan halaman admin secara umum.



Gambar 4.2 Perancangan Halaman Admin (*Dashboard*)

B. Halaman Muka (*Front-End*)

Halaman Muka atau biasa disebut *front-end*/*homepage* adalah halaman *output* dari apa yang dilakukan pada halaman admin. Website TI SMK Negeri 4 Bandung ini membagi akses pada halaman muka ini, yaitu:

* Pengunjung atau *guest* yang tidak memiliki akun hanya dapat menerima informasi dan jika ada yang ingin disampaikan bisa mengisi buku tamu.
* *User* yang memiliki akun selain dapat menerima informasi juga dipersilahkan untuk meninggalkan komentar pada artikel maupun berita.

Rancangan Halaman ini yang nantinya akan diimplementasikan pada pembuatan Website TI SMK Negeri 4 Bandung. Dibawah ini merupakan rancangan halaman muka secara umum.



Gambar 4.4 Perancangan Halaman Muka (Beranda)

Gambar 4.4 diatas merupakan penggambaran dari halaman awal muka yang akan disajikan kepada pengunjung. Sedangkan halaman artikel dan lainnya akan tampil seperti Gambar 4.5 berikut:



Gambar 4.5 Perancangan Halaman Muka | Sketsa Halaman Artikel

### Perancangan *Form*

Perancangan *form* adalah penggambaran suatu *form* yang akan dibutuhkan untuk menunjang Website TI SMK Negeri 4 Bandung. Dibawah ini akan digambarkan *form* apa saja yang akan digunakan pada pembuatan *website* ini.

1. *Form* *Login*

Untuk bisa memasuki halaman admin maupun menambahkan komentar pada halaman muka, seluruh *user* yang sudah terdaftar harus mengisi data pada *form* *login* ini. *Form* ini hanya membutuhkan Nama Pengguna dan Kata Sandi saja agar *user* yang sudah terdaftar bisa melakukan kendali penuh terhadap *website* ini. Dibawah ini adalah rancangan dari *form* *login*.



Gambar 4.4 Perancangan *Form* | *Form* Login

1. *Form* Jajak Pendapat

Untuk menambahkan suatu pemilihan pada halaman muka maka perlu menambahkan pertanyaan berikut jawabannya pada halaman admin. Pemilihan ini akan ditampilkan pada halaman muka dalam bentuk Jajak pendapat yang sebelumnya sudah ditambahkan pada *form* tambah jajak pendapat pada halaman admin. Dibawah ini akan digambarkan kedua *form* tersebut.

* 1. *Form Polling* pada *Front End*

*Form* ini dibuat untuk sekali memilih saja. Jadi tidak ada *redundancy* pada pemilihan pendapat tersebut.



Gambar 4.5 Perancangan *Form* | *Form* Jajak Pendapat pada *front-end*

* 1. *Form* Tambah Jajak Pendapat

*Form* ini dibuat untuk menambahkan pertanyaan dan jawaban untuk ditampilkan pada kolom jajak pendapat di halaman muka.



Gambar 4.6 Perancangan *Form* | *Form* Tambah Jajak Pendapat

1. *Form* Komentar

Untuk menambahkan komentar pada halaman suatu artikel atau berita maka perlu menambahkan isi komentar pada *form* komentar ini. Komentar ini akan langsung tampil dibawah *form* ini pada halaman artikel atau berita yang sebelumnya sudah ditambahkan pada *form* Komentar. Dibawah ini akan digambarkan *form* komentartersebut.



Gambar 4.7 Perancangan *Form* | *Form* Komentar

1. *Form* Buku Tamu

Untuk meninggalkan pesan, kritik, komentar maupun saran bagi *user* yang memiliki akses memberi komentar lebih mudah untuk menujukannya pada artikel maupun berita yang ingin dikomentari. Sedangkan jika *user* yang tidak memiliki akses bebas dengan komentar bisa menyampaikannya melalui *form* Buku Tamu ini. *Form* ini akan ditampilkan pada halaman muka dalam bentuk Halaman Buku Tamu yang bisa langsung diakses tanpa harus *login* terlebih dahulu. Dibawah ini akan digambarkan *form* Buku Tamu tersebut.



Gambar 4.8 Perancangan *Form* | *Form* Buku Tamu

1. *Form* Tambah Artikel

Untuk menambahkan artikel pada halaman muka semua *user* yang memiliki akses bisa menulisnya pada *form* ini. Tetapi *user* yang memiliki akses hanya sebagai siswa untuk artikel yang telah dibuatnya bisa diterbitkan pada halaman muka harus melewati penyeleksian atau persetujuan dari *user* dengan akses yang lebih berhak. Sedangkan jika *user* yang tidak memiliki akses bebas tidak bisa menulis artikel untuk diterbitkan, hanya bisa meninggalkan pesan, kritik, saran dan komentarnya pada *form* Buku Tamu(lihat penjelasan sebelumnya). Karena butuh hak akses untuk dapat menambahkan artikel, *Form* ini hanya ditampilkan pada halaman admin dalam bentuk Halaman Tambah Artikel yang bisa langsung diakses setelah *login* terlebih dahulu. Dibawah ini akan digambarkan *form* Tambah Artikel tersebut.



Gambar 4.9 Perancangan *Form* | *Form* Tambah Artikel

1. *Form* Tambah Berita
2. *Form* Tambah Kegiatan
3. *Form* Tambah Galeri
4. *Form* Tambah Majalah Dinding
5. *Form* Tambah Jajak Pendapat
6. *Form* Tambah Highlight
7. *Form* Tambah Lowongan Kerja
8. *Form* Tambah Prakerin
9. *Form* Tambah Akun
10. *Form* Sunting Akun

## Perancangan Basis Data

Perancangan basis data merupakan data-data yang dibutuhkan untuk membangun Website TI SMK Negeri 4 Bandung. Dibawah ini akan dibuat perancangan secara struktural maupun relasi.

|  |  |  |
| --- | --- | --- |
| Nama Basis Data | : | Ti\_smkn4bdg |
| Jumlah Tabel | : | 14 tabel |
| Fungsi | : | Menyimpan seluruh data yang dibutuhkan untuk pembuatan Website TI SMK Negeri 4 Bandung. |

### Struktur Tabel

Struktur tabel merupakan perincian detail data yang dibutuhkan untuk membuat suatu tabel. Dibawah ini akan dijelaskan tabel dengan *field* penunjang yang sesuai untuk membangun Website TI SMK Negeri 4 Bandung.

1. Tabel user

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | User |
| *Primary Key* | : | Username |
| *Foreign Key* | : | - |
| Fungsi | : | Menyimpan seluruh data *user* untuk mendapatkan hak akses sesuai *role*nya masing-masing. |
| Struktur | : | - |

Tabel 4.1 Tabel user

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Username | Varchar(50) | Primary Key |  |
| Password | Varchar(255) |  |  |
| Fullname | Varchar(100) |  |  |
| Avatar | Varchar(255) |  |  |
| Email | Varchar(255) |  |  |
| Bio | Text |  |  |
| Role | Enum(‘siswa’, ‘guru’, ‘admin’, ‘developer’) |  |  |

1. Tabel event

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Event |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : | Menyimpan seluruh data kegiatan yang akan ditampilkan pada Agenda di Halaman Muka. |
| Struktur | : | - |

Tabel 4.2 Tabel event

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary key | Auto\_increment |
| Title | Varchar(255) |  |  |
| From\_date | Datetime |  |  |
| Until\_date | Datetime |  |  |
| Location | Varchar(255) |  |  |
| Info | Text |  |  |
| Created\_by | Varchar(50) |  |  |
| Created\_on | Datetime |  |  |
| Views | Int(11) |  | Default: 0 |
| Status | Enum(‘archived’, ‘draft’, ‘pending’, ‘publish’) |  | Default: ‘draft’ |

1. Tabel jobs

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Jobs |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : | Menyimpan data yang berhubungan dengan penginformasian lowongan pekerjaan yang akan ditampilkan pada halaman muka. |
| Struktur | : | - |

Tabel 4.3 Tabel jobs

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary key | Auto\_increment |
| Title | Varchar(255) |  |  |
| Company | Varchar(255) |  |  |
| Logo | Varchar(255) |  |  |
| Website | Varchar(255) |  |  |
| Info | Text |  |  |
| Tags | Varchar(255) |  |  |
| Created\_by | Varchar(50) |  |  |
| Created\_on | Datetime |  |  |
| Update\_by | Varchar(50) |  |  |
| Update\_on | Datetime |  |  |

1. Tabel news

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | News |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel news

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| Title | Varchar(255) |  |  |
| Content | Text |  |  |
| Created\_by | Varchar(50) |  |  |
| Created\_on | Datetime |  |  |
| Update\_by | Varchar(50) |  |  |
| Update\_on | Datetime |  |  |
| Views | Int(11) |  | Default: 0 |
| Status | Enum(‘archived’, ‘draft’, ‘pending’, ‘publish’) |  | Default: ‘draft’ |

1. Tabel article

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Article |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel article

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | (11) | Primary Key | Auto\_increment |
| Title | Varchar(100) |  |  |
| Content | Text |  |  |
| Tags | Varchar(255) |  |  |
| Created\_by | Varchar(50) |  |  |
| Created\_on | Datetime |  |  |
| Updated\_by | Varchar(50) |  |  |
| Updated\_on | Datetime |  |  |
| Approved\_by | Varchar(50) |  |  |
| Approved\_on | Datetime |  |  |
| Views | Int(11) |  | Default: 0 |
| Likes | Text |  |  |
| Status | Enum(‘archived’, ‘draft’, ‘pending’, ‘publish’) |  | Default: ‘draft’ |

1. Tabel prakerin

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Prakerin |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel prakerin

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| Name | Varchar(255) |  |  |
| Address | Varchar(255) |  |  |
| Website | Varchar(255) |  |  |
| Category | Varchar(255) |  |  |
| Contact | Varchar(255) |  |  |
| Lat | Varchar(255) |  |  |
| Ing | Varchar(255) |  |  |
| Created\_by | Varchar(255) |  |  |
| Created\_on | Datetime |  |  |
| Updated\_by | Varchar(255) |  |  |
| Update\_on | Datetime |  |  |

1. Tabel guestbook

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Guestbook |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel guestbook

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| Name | Varchar(50) |  |  |
| Email | Varchar(50) |  |  |
| Message | Text |  |  |
| Created\_on | datetime |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Tabel highlight

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Highlight |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel highlight

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| Title | Varchar(255) |  |  |
| Link | Varchar(255) |  |  |
| Image | Varchar(255) |  |  |
| Sort | Tinyint(4) |  | Default: 0 |
| Created\_by | Varchar(50) |  |  |
| Created\_on | Datetime |  |  |

1. Tabel comments

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Comments |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel comments

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| User | Varchar(50) |  |  |
| Content | Text |  |  |
| Created\_on | Datetime |  |  |
| Likes | Text |  |  |

1. Tabel news\_comments

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | News\_comments |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel news\_comments

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| News\_id | Int(11) |  |  |
| Comment\_id | Int(11) |  |  |

1. Tabel article\_comments

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Article\_comments |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel article\_comments

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| Article\_id | Int(11) |  |  |
| Comment\_id | Int(11) |  |  |

1. Tabel tag

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Tag |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel tag

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Name | Varchar(50) | Primary Key |  |
| Frequency | Tinyint(4) |  | Default: 0 |

1. Tabel poll\_question

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Poll\_question |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel poll\_question

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| Question | Varchar(255) |  |  |
| Active | Tinyint(1) |  | Default: 0 |

1. Tabel poll\_answer

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | Poll\_answer |
| *Primary Key* | : | Id |
| *Foreign Key* | : | - |
| Fungsi | : |  |
| Struktur | : | - |

Tabel 4. Tabel poll\_answer

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| Id | Int(11) | Primary Key | Auto\_increment |
| Poll\_id | Int(11) |  |  |
| Answer | Varchar(255) |  |  |
| Total | Int(11) |  | Default: 0 |

### Relasi Tabel

(bikin kotak-kotak kasih relasi)

## Perancangan Kebutuhan Sistem

Bagian ini merupakan penjelasan mengenai perancangan kebutuhan sistem baik *hardware* maupun *software* untuk mendukung kinerja *website* ini dengan baik pada saat pengembangan maupun ketika implementasi pada perangkat komputer.

### Kebutuhan *Hardware*

Teknologi yang digunakan untuk menjalankan aplikasi dari segi kebutuhan *hardware*, dengan Spesifikasi *hardware* minimal:

|  |  |  |
| --- | --- | --- |
| *Processor* | : | Intel® PentiumIV® CPU 2.0Ghz |
| *Memory* | : | 512MB |
| *Harddisk* | : | 300MB |
| *Graphic Card* | : | 64MB |
| *Hardware* lain | : | Ethernet Card dengan kecepatan 10/100Mbps (untuk penggunaan dalam jaringan) |

### Kebutuhan *Software*

Teknologi yang digunakan untuk menjalankan aplikasi dari segi kebutuhan *software* dengan Spesifikasi *software* sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Sistem Operasi | : | Windows XP Professional SP2 ke atas, Linux, Mac OSX |
| Bahasa Pemrograman | : | PHP Hypertext Preprocessor (5 ke atas) |
| Paket Instalasi Lain | : | XAMPP 1.7.x (1.7.0 ke atas direkomendasikan, berisi paket instalasi web server) |
| *Software* lain | : | Web Browser |

## Jadwal Pembuatan Aplikasi

Dalam pengerjaan Website TI SMK Negeri 4 Bandung ini penulis membutuhkan beberapa perencanaan dan agenda yang telah tersusun di dalam Gantt chart berikut :

Tabel 4. *Gantt Chart*



# implementasi

Pada bab ini akan menguraikan tahapan implementasi dari perancangan dan analisis masalah yang telah dibuat dan dijelaskan pada bab sebelumnya. Bab ini akan dibagi kedalam 4 bagian implementasi, yaitu perangkat implementasi, implementasi peta situs (*site map*), antar muka sistem, dan basis data.

## Perangkat Implementasi

Pada bab sebelumnya telah dijelaskan perancangan kebutuhan sistem. Dari perancangan itulah beberapa perangkat implementasi digunakan untuk membuat dan menyempurnakan Website TI SMK Negeri 4 Bandung. Untuk proses pembuatan Website TI SMK Negeri 4 Bandung diimplementasikan menggunakan Netbeans IDE, dengan MySQL sebagai *server* basis datanya dan PHP sebagai salah satu bahasa pemrogramannya.

Dalam semua proses yang akan menghasilkan suatu *website,* perangkat pengujian sangat penting untuk didefinisikan dan dibatasi sejak diawal pembuatan dan pengembangan *website* ini. Dengan menyesuaikan dengan perancangan kebutuhan sistem, maka pada pengujian Website TI SMK Negeri 4 Bandung ini menggunakan web browser yang *support* terhadap segala fasilitas yang ada. Beberapa web browser yang disarankan adalah Firefox, Chrome, Safari, dan browser terbaru lainnya direkomendasikan. Untuk IE, IE8.0 tidak direkomendasikan.

## Implementasi Peta Situs (*Site Map*)

Berdasarkan perancangan Peta Situs yang sudah dijelaskan pada bab sebelumnya, pada peta situs terdapat beberapa menu utama yang akan langsung tampil pada halaman utama Website TI SMK Negeri 4 Bandung, beberapa diantaranya adalah Berita Terbaru, Sorotan, Artikel, dan Rubrik Edukasi. Dibawah ini akan ditampilkan implementasi peta situs Website TI SMK Negeri 4 Bandung.

Gambar 5.1 Implementasi Menu Awal

Tampilan diatas tersebut merupakan implementasi menu awal atau biasa dikenal dengan *homepage* dari situs. Untuk menampilkan halaman tersebut, dibuatlah beberapa *source code*. Dalam Zend Framework, untuk membuat sebuah tampilan seperti diatas, dibutuhkan *model*, *view*,dan *controller*. Tapi karena pada Menu Awal tidak ada data yang langsung mengakses pada basis data, maka *model* untuk tampilan ini tidak ada. Berikut ini adalah potongan dari *view* dan *controller* tersebut:

|  |
| --- |
| **View (index.phtml)** |
|  |

|  |
| --- |
| **Controller (indexController.php)** |
|  |

## Implementasi Antar Muka Sistem

Implementasi antar muka sistem adalah perwujudan dari perancangan yang telah disusun menjadi lebih nyata. Subbab ini akan dibagi lagi menjadi beberapa bagian implementasi yang sudah dikelompokkan sesuai fungsi agar lebih mudah dipahami. Dibawah ini akan disertakan implementasi-implementasi tersebut dalam bentuk visual agar lebih mudah tergambarkan.

### Tampilan Halaman

Implementasi tampilan halaman adalah bagian perwujudan dari proses perancangan yang sudah digambarkan pada bab sebelumnya sehingga lebih sederhana untuk diimplementasikan. Dalam hal ini implementasi tampilan halaman dibuat seperti pada perancangan, dari dua sudut pandang yang berbeda. Dibawah ini akan disertakan gambar dua halaman tersebut dalam fokus fungsi berbeda.

1. Halaman Admin

Seperti pada penjelasan bab sebelumnya, halaman admin dapat diakses jika sudah terdaftar sebagai *user* dengan berbagai jenis *role* yang berbeda. Dibawah ini akan disertakan gambar dari implementasi halaman admin dan beberapa perbedaan fasilitas pada akses *user*.

Gambar 5.2 Implementasi Tampilan Halaman Admin

Tampilan diatas tersebut merupakan implementasi tampilan halaman admin atau biasa dikenal dengan istilah *back-end* dari situs. Untuk menampilkan halaman tersebut, dibuatlah beberapa *source code*. Berikut ini adalah potongan dari *source code* tersebut:

|  |
| --- |
| **Index.phtml** |
|  |

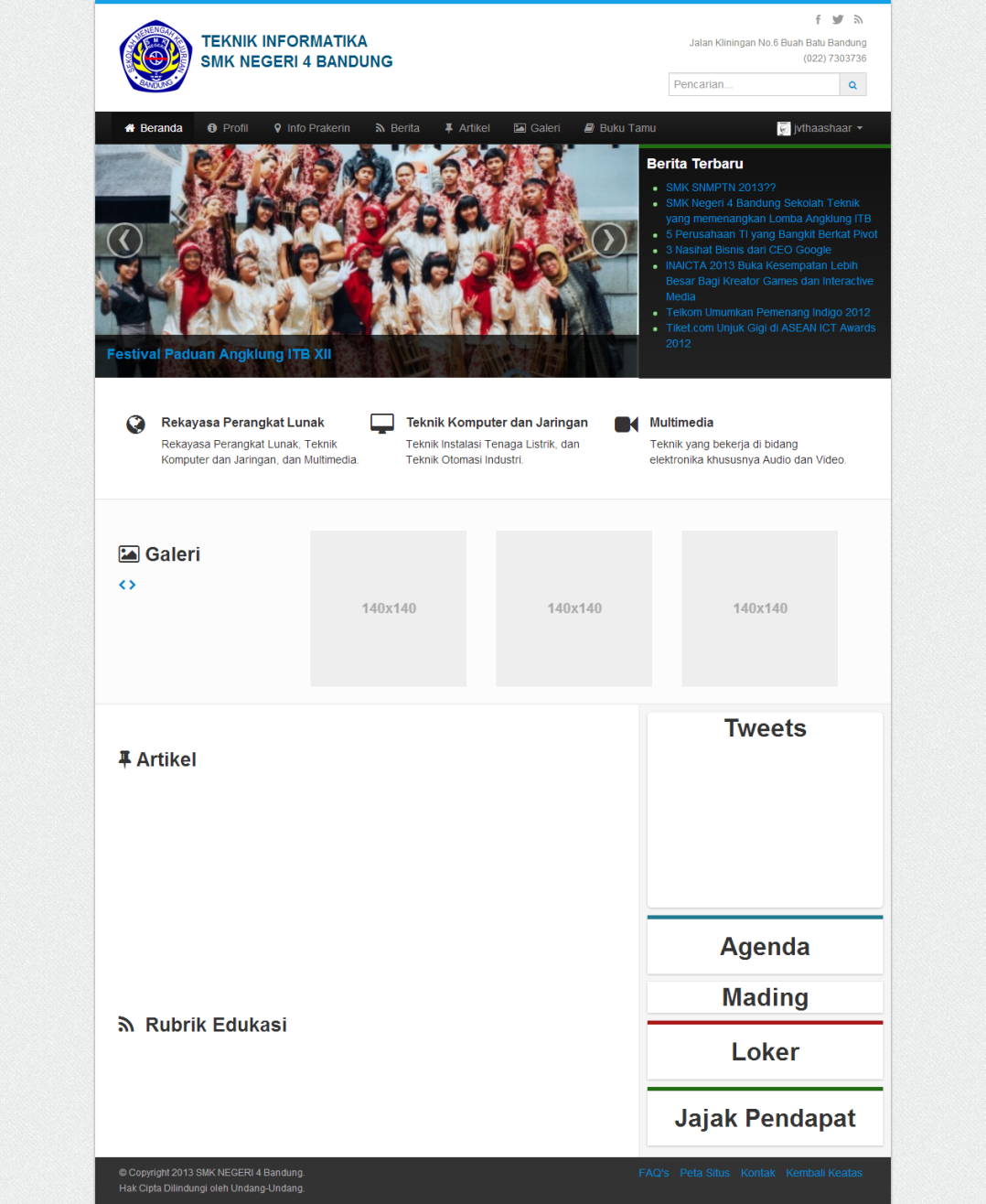
Dibawah ini akan dijelaskan perbedaan tampilan halaman admin sesuai hak akses setiap *user*.

Tabel 5.1 Perbedaan Halaman Admin (sesuai *role user*)

|  |  |  |  |
| --- | --- | --- | --- |
| ***Role*** | **Dashboard** | **Konten** | **Admin** |
| Siswa | Pada dashboard tidak ada *listing* buku tamu. | Pada Konten hanya dapat mengakses menu artikel. | Tidak memiliki hak akses atur admin. |
| Guru | Pada dashboard tidak ada *listing* buku tamu. | Pada Konten hanya dapat mengakses menu artikel, berita, dan galeri. | Pada Admin hanya dapat mengakses menu diatas. |
| Admin | Pada dashboard terdapat *listing* buku tamu dan komentar. | Pada Konten dapat mengakses seluruh menu diatas. | Pada Admin dapat mengakses seluruh menu diatas. |

1. Halaman Muka

Seperti pada penjelasan bab sebelumnya, halaman muka dapat diakses oleh siapa saja. Yang membedakan hanya pada pengguna yang tidak memiliki *user* tidak dapat meninggalkan komentar, hanya bisa mengakses buku tamu saja. Dibawah ini akan disertakan gambar dari implementasi halaman muka.



Gambar 5.3 Implementasi Tampilan Halaman Muka

Tampilan diatas tersebut merupakan implementasi tampilan halaman muka atau biasa dikenal dengan istilah *homepage*  dari situs. Untuk menampilkan halaman tersebut, dibuatlah beberapa *source code*. Berikut ini adalah potongan dari *source code* tersebut:

|  |
| --- |
| **layout.phtml** |
|  |

Gambar 5.3 diatas merupakan penggambaran dari halaman awal yang akan disajikan kepada pengunjung. Sedangkan halaman artikel dan lainnya akan tampil seperti Gambar 5.4 berikut:

Gambar 5.4 Implementasi Tampilan Halaman Muka | Konten: Artikel

Tampilan diatas tersebut merupakan implementasi tampilan halaman muka konten artikel dari Website TI SMK Negeri 4 Bandung. Untuk menampilkan halaman tersebut, dibuatlah beberapa *source code*. Berikut ini adalah potongan dari *source code* tersebut:

|  |
| --- |
| **ArticleController.phtml** |
|  |

### Tampilan *Form*

Implementasi tampilan *form* adalah bagian perwujudan dari suatu *form* yang sudah dirancang pada bab sebelumnya. Dibawah ini akan digambarkan *form* apa saja yang akan digunakan pada pembuatan *website* ini.

1. *Form Login*
2. *Form* Jajak Pendapat
   1. *Form Polling* pada *Front End*
   2. *Form* Tambah Jajak Pendapat
3. *Form* Komentar
4. *Form* Buku Tamu
5. *Form* Tambah Aritkel
6. *Form* Tambah Berita
7. *Form* Tambah Kegiatan
8. *Form* Tambah Galeri
9. *Form* Tambah Majalah Dinding
10. *Form* Tambah Jajak Pendapat
11. *Form* Tambah Highlight
12. *Form* Tambah Lowongan Kerja
13. *Form* Tambah Prakerin
14. *Form* Tambah Akun
15. *Form* Sunting Akun

## Implementasi Basis Data

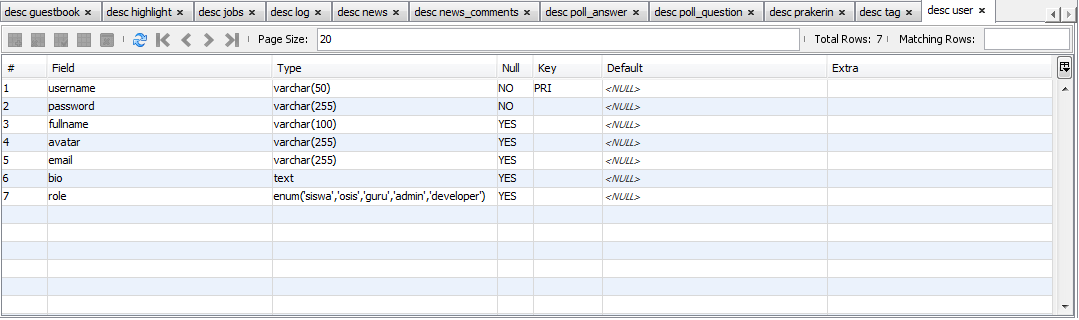
(prtsc itu database ☺)

### Struktur Tabel

(pertabel menggunakan numbering)

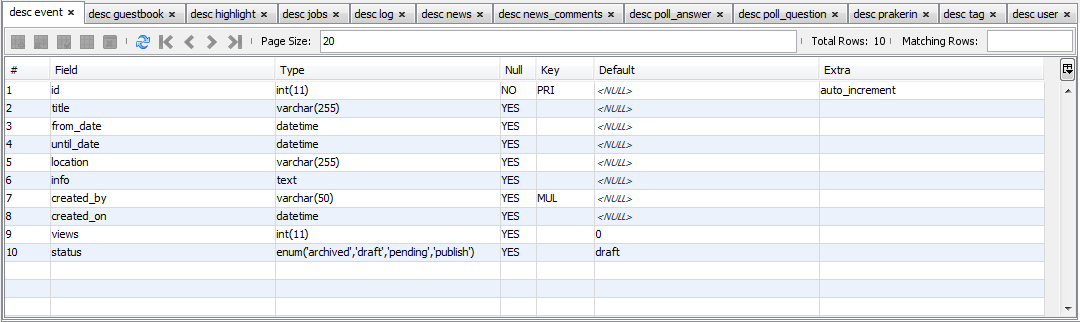
1. Tabel user

Tabel 5. Tabel user



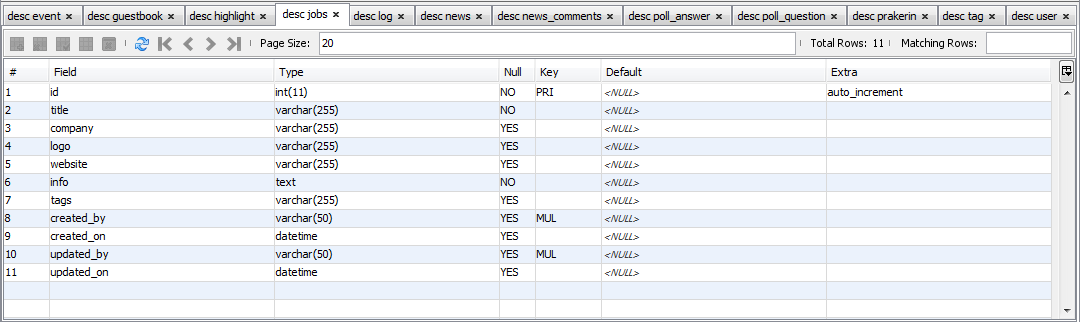
1. Tabel event

Tabel 5. Tabel event



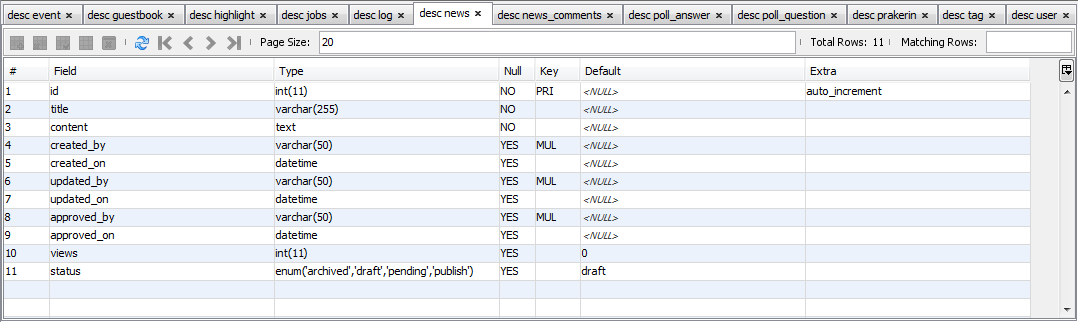
1. Tabel jobs

Tabel 5. Tabel jobs



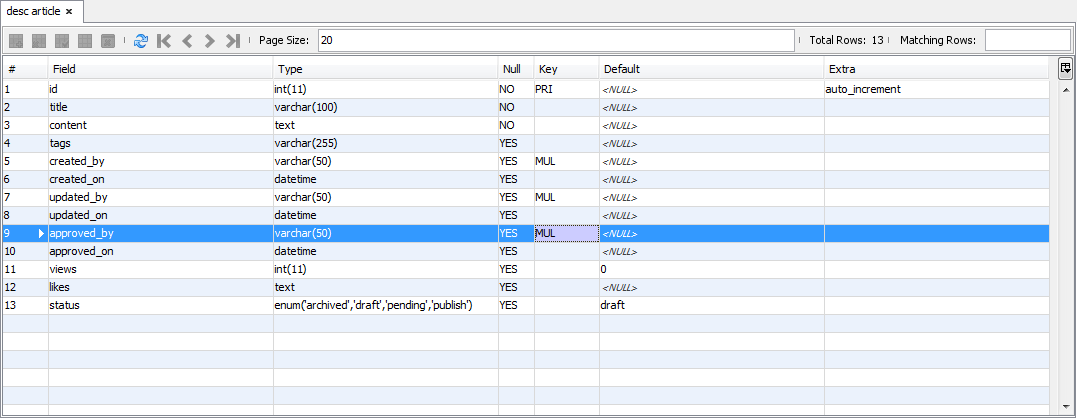
1. Tabel news

Tabel 5. Tabel news



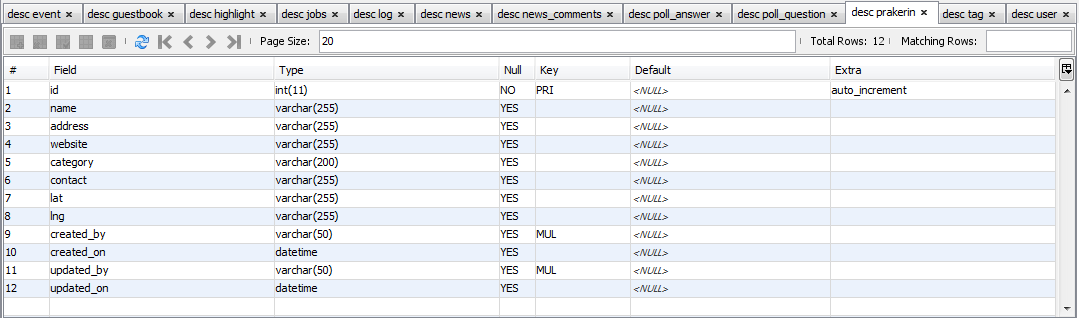
1. Tabel article

Tabel 5. Tabel article



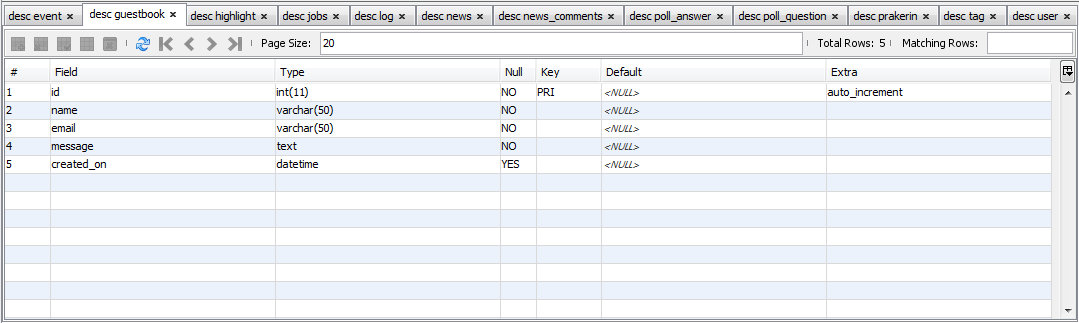
1. Tabel prakerin

Tabel 5. Tabel prakerin



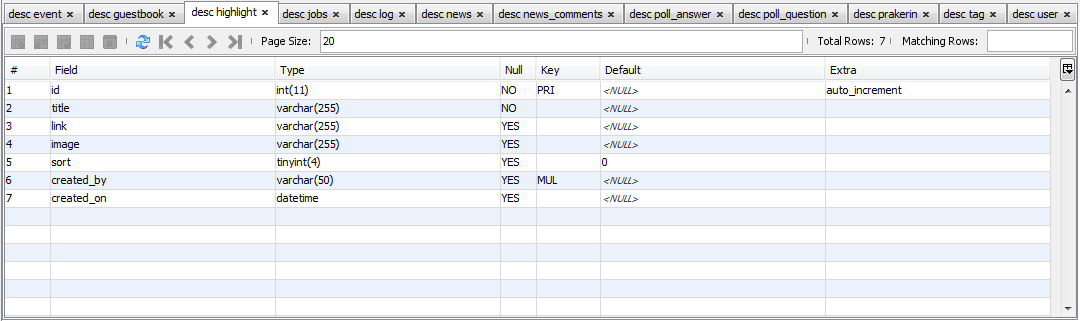
1. Tabel guestbook

Tabel 5. Tabel guestbook



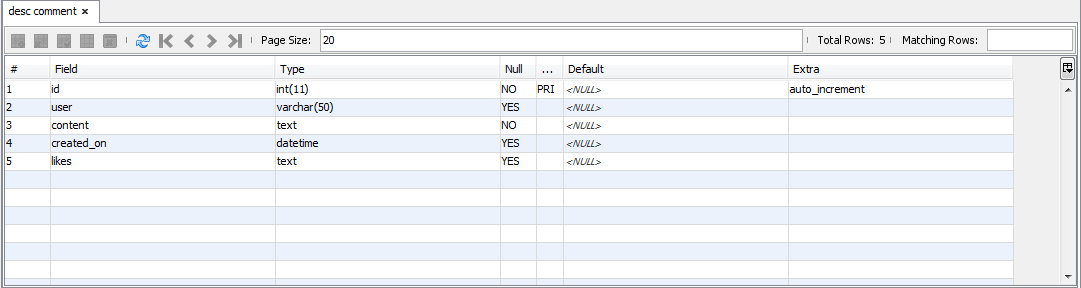
1. Tabel highlight

Tabel 5. Tabel highlight



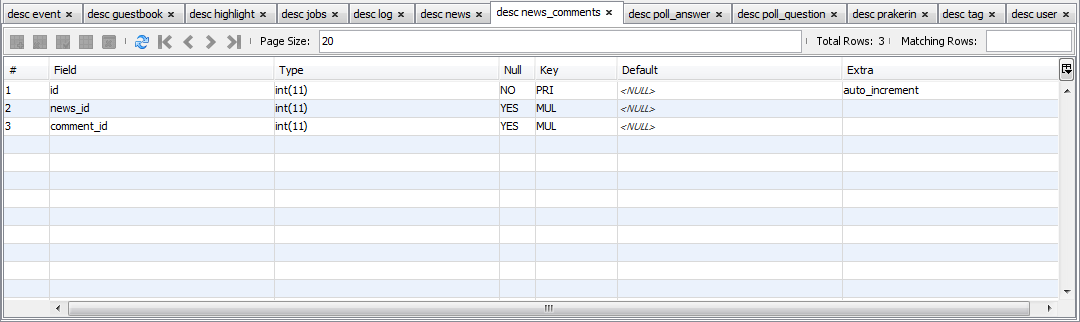
1. Tabel comments

Tabel 5. Tabel comments



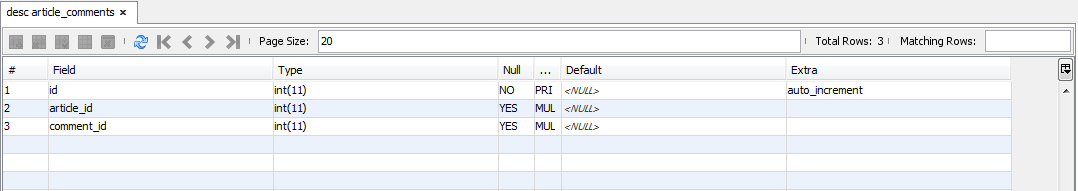
1. Tabel news\_comments

Tabel 5. Tabel news\_comments



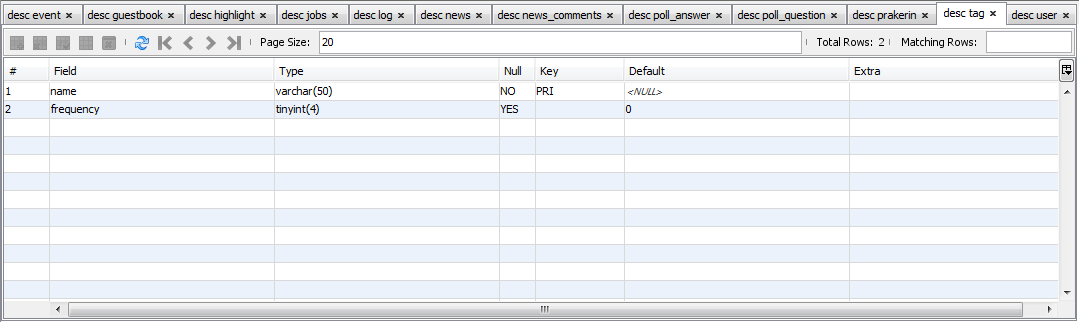
1. Tabel article\_comments

Tabel 5. Tabel article\_comments



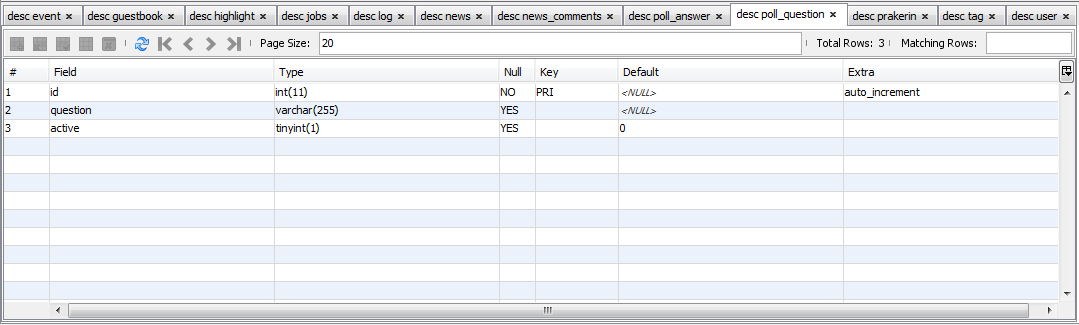
1. Tabel tag

Tabel 5. Tabel tag



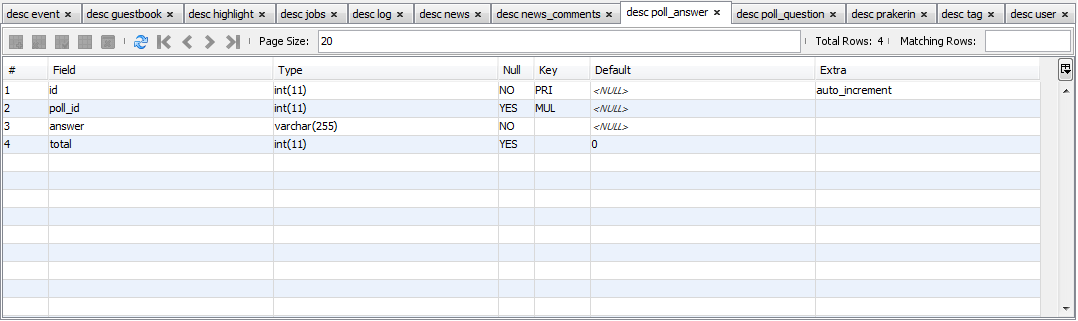
1. Tabel poll\_question

Tabel 5. Tabel poll\_question



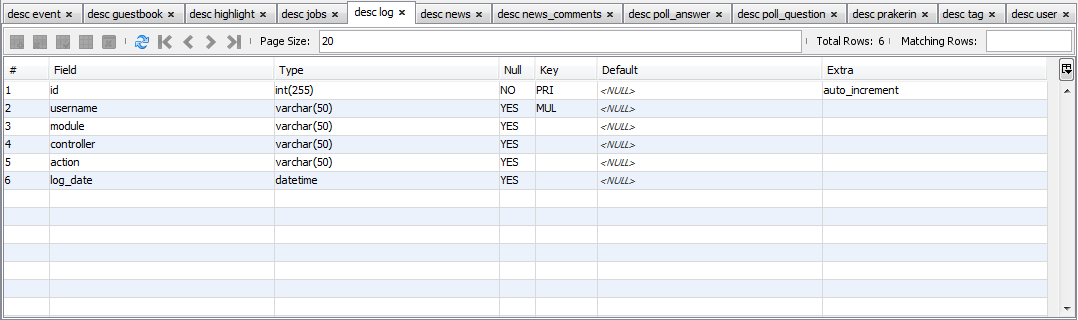
1. Tabel poll\_answer

Tabel 5. Tabel poll\_answer



1. Tabel log (delete)

Tabel 5. Tabel log



### Relasi Tabel

# penutup

## Kesimpulan

## Saran Pengembangan

# daftar pustaka

Hakim, L., and U. Musalini. *Cara Mudah Memadukan Web Design dan Web Programming.* Jakarta: PT. Elex Media Komputindo, 2004.

Javan Cipta Solusi. *Apa itu Zend Framework??* 16 Maret 2012. http://javan.co.id/apa-itu-zend-framework/ (diakses April 2, 2013).

Kadir, Abdul. *Pemrograman Web Mencakup HTML, CSS, JavaScript dan PHP.* Jogjakarta: Andi Offset, 2003.

Kahfi, Seto El. *Apa itu Framework PHP??* 4 Januari 2013. http://blog.setoelkahfi.web.id/2013/01/Apa-Itu-Framework-PHP-Pengertian-Framework-PHP.html (diakses April 2, 2013).

Peranginangin, Kasiman. *Aplikasi Web dengan PHP dan MySql.* Jogjakarta: Andi Offset, 2006.

PRmob.net. *Zend Framework dan Keuntungannya.* t.thn. http://id.prmob.net/zend-framework/php/model-view-controller-1955043.html (diakses April 2, 2013).

Thaashaar, Julyestra Vidha. *APLIKASI PEMBELAJARAN SANDI MORSE BERBASIS J2ME.* Bandung: SMK Negeri 4 Bandung, 2012.

Zend Technologies Ltd. *Zend Framework.* 2006-2013. http://framework.zend.com/ (diakses April 2, 2012).