# SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER JAKARTA STI&K



**SKRIPSI**

|  |
| --- |
| **RANCANG BANGUN APLIKASI MEDIA PEMBELAJARAN CRUD JAVA WEB DENGAN MENGGUNAKAN NETBEANS 8.2 DAN MYSQL**  **Nama : Nabil Fawwaz Elqayyim**  **NPM : 10916274**  **Program Studi : Sistem Informasi**  **Dosen Pembimbing : Fitri Sjafrina, SKom., MMSI.** |

**Diajukan Guna Melengkapi Sebagai Syarat Untuk**

**Mencapai Jenjang Strata Satu (S1**)

**JAKARTA**

**2017**

# LEMBAR PENGESAHAN

Nama : Nabil Fawwaz Elqayyim

NPM : 10916274

Program Studi : S1 – Sistem Informasi

Judul : Rancang Bangun Aplikasi Media Pembelajaran CRUD Java Web Menggunakan Netbeans 8.2 dan MySQL

Dosen Pembimbing : Fitri Sjafrina, SKom., MMSI.

Tanggal Sidang : 8 November 2017

Tanggal Lulus : 8 November 2017

Dosen Penguji : Dr. Pipit Dewi Arnesia

Dr. Laode M. Rasdi Rere

Menyetujui:

**Dosen Pembimbing Koordinator Sidang**

(Fitri Sjafrina, Skom., MMSI.) (Diyah Ruri Irawati, Skom., MMSI.)

**Ketua Program Studi Sistem Informasi**

(Dr. Pipit Dewi Arnesia)

# PERNYATAAN KARYA ASLI DAN PUBLIKASI

Saya yang bertanda tangan pernyataan dibawah ini:

Nama : Nabil Fawwaz Elqayyim

NPM : 10916274

Judul Penelitian Penulisan : Rancang Bangun Aplikasi Media Pembelajaran CRUD Java Web Menggunakan Netbeans 8.2 dan MySQL

Tanggal Sidang : 8 November 2017

Tanggal Lulus : 8 November 2017

Menyatakan bahwa tulisan ini adalah karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh STMIK Jakarta STI&K. Segala kutipan dalam bentuk apapun telah mengikuti kaidah, dan etika yang berlaku. Mengenai isi dan tulisan adalah merupakan tanggung jawab Penulis, bukan STMIK Jakarta STI&K.

Demikian pernyataan ini dibuat sebenarnya dan dengan penuh kesadaran.

Jakarta, 6 November 2017

(Nabil Fawwaz Elqayyim)

# ABSTRAK

**Rancang Bangun Aplikasi Media Pembelajaran CRUD Java Web Menggunakan Netbeans 8.2 Dan MySQL**

Nabil Fawwaz Elqayyim, 10916274

Skripsi, Sistem Informasi, 2017

STMIK Jakarta STI&K

<http://www.jak-stick.ac.id>

Kata Kunci : Aplikasi, CRUD, Java Web, Netbeans 8.2

( xii + 42 Hal + Lampiran )

Media pembelajaran secara umum adalah alat bantu proses belajar mengajar. Segala sesuatu yang dapat dipergunakan untuk merangsang pikiran, perasaan, perhatian dan kemampuan atau keterampilan pebelajar sehingga dapat mendorong terjadinya proses belajar. Banyak hal bisa didapatkan melalui media pembelajaran lewat komputer ini, salah satunya adalah belajar bahasa pemrograman. Dari banyaknya bahasa pemrograman yang ada, bahasa pemrograman yang saat ini sedang populer adalah bahasa pemrograman Java. Media pembelajaran dengan menggunakan bahasa pemrograman Java Web masih terbilang sedikit di internet, perlu menggunakan kata kunci bahasa inggris di mesin pencarian untuk menemukan tutorial bahasa pemrograman Java Web. Banyaknya orang-orang yang berminat belajar bahasa pemrograman Java tapi kurang mengerti dari tutorial yang mereka pelajari dari internet. Secara umum, CRUD sering digunakan sebagai dasar untuk pembuatan aplikasi sederhana, dengan kata lain, CRUD juga bisa dijadikan bahan media pembelajaran dasar bahasa pemrograman. Sebagai media pembelajaran dasar, biodata karyawan sangat tepat untuk dijadikan objek yang akan diimplementasikan kedalam aplikasi CRUD. Dalam kasus ini, peneliti menggunakan biodata karyawan PT. AITINDO sebagai objek yang akan diimplementasikan kedalam aplikasi CRUD dengan bahasa pemrograman Java berbasis *web*.

Dosen Pembimbing : Fitri Sjafrina SKom., MSSI.

Daftar Pustaka : 16, (2002-2017)

# ABSTRACT

**Design Build Learning Media Application CRUD Java Web Using Netbeans 8.2 And MySQL**

Nabil Fawwaz Elqayyim, 10916274

Minithesis, Information System, 2017

STMIK Jakarta STI&K

<http://www.jak-stick.ac.id>

Keyword : Application, CRUD, Java Web, Netbeans 8.2

( xii + 42 Pages + Attachments )

Learning media in general is a tool of teaching and learning process. Everything that can be used to create thoughts, feelings, concerns and skills. Many things can be obtained through learning media through this computer, one of which is learning programming language. Of the many programming languages ​​that exist, the programming language that is currently popular is the Java programming language. Learning media using the Java Web programming language is still fairly little on the internet, need to use the English keyword in the search engine to find the Java Web programming language tutorial. The number of people who are interested in learning Java programming language but less understanding of the tutorials that they read from the internet. In general, CRUD is often used as a basis for making simple applications, in other words, CRUD can also be used as a basic instructional media for programming languages. As a basic learning media, employees biodata is very appropriate to be used as objects that will be implemented in CRUD applications. In this case, researchers use biodata employees of PT. AITINDO as an object to be implemented in CRUD application with java web-based programming language.

Supervisor : Fitri Sjafrina SKom., MSSI.

Bibliography : 16, (2002-2017)

# KATA PENGANTAR

Assalamu’alaikum Warrahmatullahi Wabarakatuh.

Dengan memanjatkan puji serta syukur kehadirat Allah Subhanahu Wata’ala, atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penulisan ilmiah ini. Shalawat serta salam semoga Allah Subhanahu Wata’ala selalu melimpahkan kepada Baginda Rasullah Muhammad Shallallahu ‘Alaihi Wasallam beserta keluarga, sahabat dan umatnya hingga akhir zaman yang tetap setia mengikuti beliau.

Penulis menyadari bahwa penulisan ilmiah ini masih banyak kekurangan dan jauh dari kesempurnaan. Meskipun demikian penulis berharap semoga penulisan ilmiah ini bermanfaat bagi semua pihak yang membutuhkan.

Pada kesempatan ini penulis tak lupa menyampaikan ucapan terima kasih yang setinggi-tingginya atas bantuan dan bimbingan yang penulis terima. Oleh karena ssitu penulis dalam kesempatan ini ingin mengucapkan banyak terima kasih kepada:

1. Ibu Dr. Lussiana, ETP, selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer JAKARTA STI&K.
2. Ibu Dr. Pipit Dewi Arnesia, selaku Ketua Program Studi Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer JAKARTA STI&K.
3. Ibu Diyah Ruri Irawati, SKom., MMSI., selaku Koordinator sidang.
4. Ibu Fitri Sjafrina, SKom., MMSI., selaku Dosen Pembimbing yang telah memberikan perhatian, bimbingan, arahan, serta saran kepada penulis dalam menyusun penulisan ilmiah ini dari mulai persiapan hingga selesainya penulisan ilmiah ini .
5. Seluruh Bapak dan Ibu Dosen yang berada di lingkungan Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K, yang telah memberikan ilmu serta motivasi kepada penulis.
6. Kelembagaan STMIK Jakarta STI&K BPM, BEM dan Keluarga Besar Komunitas Pencinta Komputer STMIK Jakarta STI&K (KOMPUTIKA), serta teman-teman dekat saya yang telah memberikan do’a, ilmu, motivasi, dukungan dan semangatnya.
7. Semua teman-teman saya, dari mulai teman di rumah, SD, SMP, SMA sampai teman saya di perkuliahan ini dan kakak-kakak senior saya yang selalu saling memberikan motivasi agar selalu semangat untuk saling mendukung satu sama lain agar terselesaikannya penulisan ini.

Ucapan terima kasih yang tiada tara untuk kedua orang tua penulis. Untuk Ibu dan Ayah yang telah menjadi orang tua terhebat sejagad raya, yang selalu memberikan motivasi, nasehat, cinta, perhatian, kasih sayang serta doa dan pengorbanannya baik dari segi moril, materi kepada penulis semenjak kecil sehingga penulis dapat menyelesaikan penulisan ini yang tentu takkan pernah bisa penulis balas. Penulis menyadari bahwa dalam penulisan skripsi ini tentunya masih jauh dari kata “sempurna”. Pasti masih banyak kekurangan, baik berupa kata, uraian, maupun pembahasannya, hal ini karena keterbatasan penulis dalam hal kemampuan dan pengetahuan. Untuk itu penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Akhir kata penulis berharap semoga penulisan ini dapat bermanfaat bagi semua pihak terutama yang membutuhkannya. Amin.

Wassalamu’alaikum warahhmatulahi wabarakatuh.

Jakarta, 6 November 2017

Penulis

# DAFTAR ISI

[JUDUL HALAMAN i](#_Toc501100728)

[LEMBAR PENGESAHAN ii](#_Toc501100729)

[PERNYATAAN KARYA ASLI DAN PUBLIKASI iii](#_Toc501100730)

[ABSTRAK iv](#_Toc501100731)

[ABSTRACT iv](#_Toc501100732)

[KATA PENGANTAR vi](#_Toc501100733)

[DAFTAR ISI viii](#_Toc501100734)

[DAFTAR GAMBAR x](#_Toc501100735)

[DAFTAR TABEL xi](#_Toc501100736)

[1. PENDAHULAN 1](#_Toc501100737)

[1.1 Latar Belakang Masalah 1](#_Toc501100738)

[1.2 Rumusan Masalah dan Batasan Masalah 2](#_Toc501100739)

[1.3 Tujuan Penelitian 2](#_Toc501100740)

[1.4 Metode Penelitian 3](#_Toc501100741)

[1.5 Sistematika Penulisan 7](#_Toc501100742)

[2. TINJAUAN PUSTAKA 9](#_Toc501100743)

[2.1 Aplikasi 9](#_Toc501100744)

[2.2 Website 9](#_Toc501100745)

[2.2.1 Jenis-Jenis Website 10](#_Toc501100746)

[2.2.2 Fungsi-Fungsi Website 11](#_Toc501100747)

[2.2.3 Manfaat Website 11](#_Toc501100748)

[2.2.4 HTML 12](#_Toc501100749)

[2.2.5 XML 13](#_Toc501100750)

[2.2.6 Javascript 14](#_Toc501100751)

[2.2.7 Bootstrap 15](#_Toc501100752)

[2.3 Java 16](#_Toc501100753)

[2.3.1 Kelebihan Dan Kekurangan Java 16](#_Toc501100754)

[2.3.2 JDK (Java Development Kit) 18](#_Toc501100755)

[2.3.3 JRE (Java Runtime Environment) 19](#_Toc501100756)

[2.3.4 IDE (Integrated Development Environment) 19](#_Toc501100757)

[2.3.5 JVM (Java Virtual Machine) 19](#_Toc501100758)

[2.3.6 JSP Dan Servlet 19](#_Toc501100759)

[2.4 Netbeans 20](#_Toc501100760)

[2.4.1 JSE, JEE, JME Pada Netbeans 21](#_Toc501100761)

[2.4.2 Kelebihan dan Kekurangan Netbeans 22](#_Toc501100762)

[2.5 Database 23](#_Toc501100763)

[2.5.1 DBMS 23](#_Toc501100764)

[2.5.2 Macam-Macam DBMS 24](#_Toc501100765)

[2.5.3 Keuntungan Dan Kekurangan DBMS 26](#_Toc501100766)

[2.5.4 Fasilitas DBMS 27](#_Toc501100767)

[2.5.5 RDBMS 28](#_Toc501100768)

[2.5.6 JDBC 28](#_Toc501100769)

[2.6 Maven 30](#_Toc501100770)

[2.7 CRUD 30](#_Toc501100771)

[3. ANALISA DAN PERANCANGAN APLIKASI 32](#_Toc501100772)

[3.1 Struktur Navigasi 32](#_Toc501100773)

[3.2 Rancangan Database 33](#_Toc501100774)

[3.3 Rancangan Halaman Utama 35](#_Toc501100775)

[3.4 Rancangan Halaman Tambah Data 36](#_Toc501100776)

[3.5 Rancangan Halaman Ubah Data 37](#_Toc501100777)

[4. HASIL IMPLEMENTASI DAN UJICOBA APLIKASI 38](#_Toc501100778)

[4.1 Tampilan Halaman Utama 38](#_Toc501100779)

[4.2 Tampilan Halaman Tambah Data 39](#_Toc501100780)

[4.3 Tampilan Halaman Ubah Data 40](#_Toc501100781)

[5. KESIMPULAN DAN SARAN 41](#_Toc501100782)

[5.1 Kesimpulan 41](#_Toc501100783)

[5.2 Saran 41](#_Toc501100784)

[DAFTAR PUSTAKA 42](#_Toc501100785)

[LAMPIRAN L-xliii](#_Toc501100786)

# DAFTAR GAMBAR

No. Teks Hal

1.1 Kuisioner 1 3

1.2 Kuisioner 2 4

1.3 Kuisioner 3 5

1.4 Kuisioner 4 5

1.5 Waterfall 6

2.1 Logo Aplikasi 9

2.2 Logo Website 9

2.3 Skema Kerja HTML 10

2.4 Logo HTML 12

2.5 Logo Javascript 14

2.6 Logo Bootstrap 15

2.7 Logo Java 16

2.8 Logo Netbeans 20

2.9 Logo Database 23

3.1 Struktur Navigasi 33

3.2 Rancangan Halaman Utama 35

3.3 Rancangan Halaman Tambah Data 36

3.4 Rancangan Halaman Ubah Data 37

4.1 Tampilan Halaman Utama 38

4.2 Tampilan Halaman Tambah Data 39

4.2 Tampilan Halaman Ubah Data 40

# DAFTAR TABEL

No. Teks Hal

3.1 Biodata 34

# 1. PENDAHULAN

## 1.1 Latar Belakang Masalah

Media pembelajaran merupakan alat bantu proses belajar mengajar. Segala sesuatu yang dapat dipergunakan untuk merangsang pikiran, perasaan, perhatian dan kemampuan atau keterampilan orang yang belajar, sehingga dapat mendorong terjadinya proses belajar yang mencakup sumber, lingkungan, manusia dan metode yang dimanfaatkan untuk tujuan pembelajaran/pelatihan. Sekarang ini, media pembelajaran lewat komputer menjadi salah satu yang paling sering digunakan. Banyak hal bisa didapatkan melalui media pembelajaran lewat komputer ini, salah satunya adalah belajar bahasa pemrograman.

Dari banyaknya bahasa pemrograman yang ada, bahasa pemrograman yang saat ini sedang populer adalah bahasa pemrograman Java. Banyak sekali tutorial-tutorial bahasa pemrograman Java yang bisa didapatkan di internet, baik berupa artikel maupun video. Bahasa pemrograman Java pun banyak jenis *platform*nya. Seperti *Java Standard Edition*(Java Desktop), *Java Micro Edition*(Java Mobile), dan *Java Enterprise Edition*(Java Web). Dari 3 platform Java diatas, *Java Enterprise Edition* atau bisa dibilang Java Web lah yang terbilang tutorialnya sangat sedikit di internet. Orang-orang cenderung perlu menggunakan bahasa Inggris sebagai kata kunci untuk mencari sebuah tutorial Java Web di mesin pencarian jika tidak menemukan tutorial tersebut dalam bahasa Indonesia. Inilah masalah yang sedang dihadapi bagi orang-orang yang ingin belajar bahasa pemrograman Java Web.

Saat ini, teknik pengolahan data bisa menjadi objek belajar dan menjadi bagian yang tidak terpisahkan dari awal proses belajar mengajar bahasa pemrograman demi tercapainya tujuan pendidikan pada umumnya dan tujuan pembelajaran di perguruan tinggi pada khususnya. Banyak mahasiswa yang ingin belajar bahasa pemrograman Java Web karena ini adalah salah satu bahasa pemrograman yang sekarang ini sedang populer. Objek yang digunakan sebagai pengolahan data menggunakan biodata karyawan PT. AITINDO,[Lampiran].

Berdasarkan uraian yang telah dijabarkan diatas, maka peneliti membuat rancang bangun aplikasi media pembelajaran CRUD dengan bahasa pemrograman Java berbasis web sampai dengan implementasi dan hasil uji coba rancang bangun aplikasi yang dibuat dengan Netbeans 8.2.

## 1.2 Rumusan Masalah dan Batasan Masalah

Melihat permasalahan yang terdapat dalam latar belakang masalah, ditemukan beberapa masalah yang dapat dirumuskan dalam suatu rumusan masalah adalah sebagai berikut:

1. Media pembelajaran dengan menggunakan bahasa pemrograman Java Web masih terbilang sedikit di internet, perlu menggunakan kata kunci bahasa inggris di mesin pencarian untuk menemukan tutorial bahasa pemrograman Java Web.
2. Banyaknya orang-orang yang berminat belajar bahasa pemrograman Java tapi kurang mengerti dari tutorial yang mereka pelajari dari internet.

Dari rumusan masalah yang telah diuraikan diatas, dapat diambil batasan masalah yang ada pada penelitian ini adalah sebagai berikut:

1. Biodata karyawan yang sebagai objek pada rancang bangun aplikasi ini hanya menggunakan 8 buah objek data.
2. Aplikasi dibuat dengan skala kecil, karena ditujukan untuk pembalajaran awal saja.

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah**:**

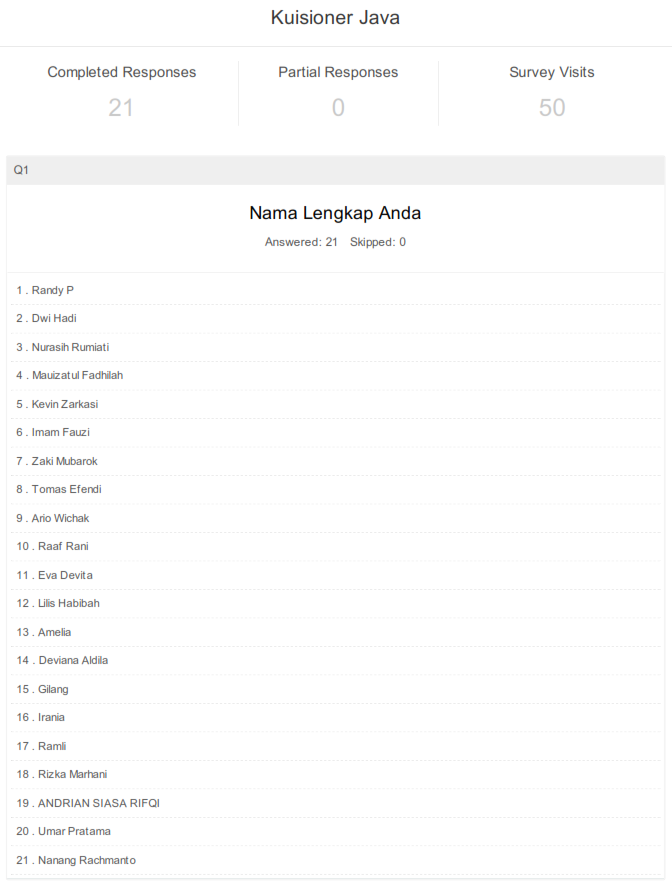
1. Menghasilkan aplikasi pengolahan data menggunakan metode CRUD dengan bahasa pemrograman Java berbasis web sebagai media pembalajaran dasar.
2. Memudahkan orang-orang untuk belajar dasar bahasa pemrograman Java berbasis web.

## 1.4 Metode Penelitian

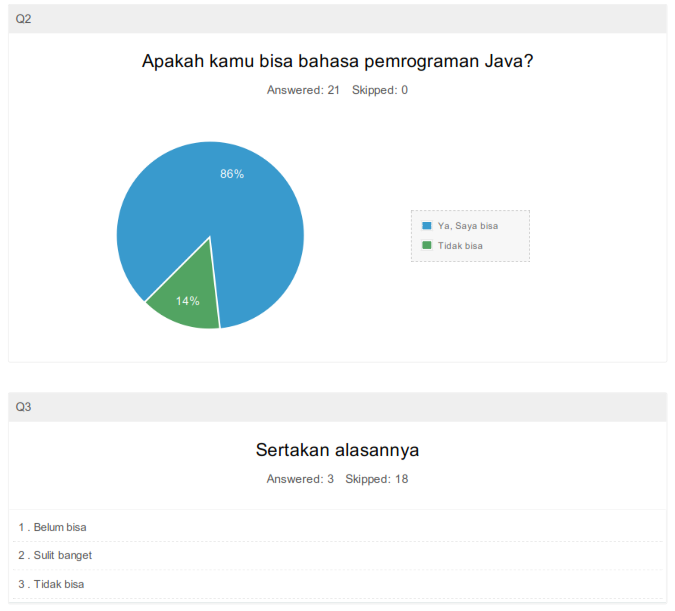
Untuk penyelesaian penelitian ini diperlukan pengumpulan data yang berhubungan dengan masalah yang dibahas. Tujuannya adalah sebagai sumber landasan pembahasan dan pembuatan rancangan aplikasi. Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Tahap pengumpulan data yang digunakan dalam penelitian adalah :
2. Metode Pengumpulan Data

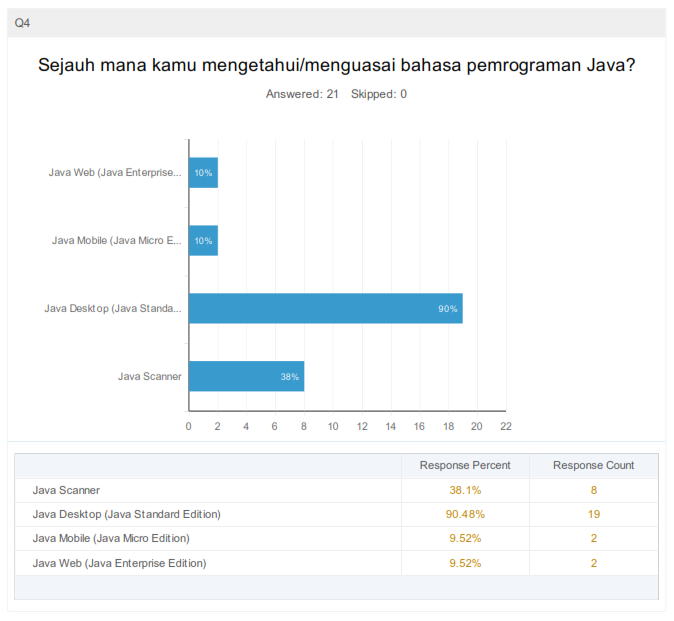
Pengumpulan data dengan cara membuat kuisioner online dan diberikan kepada orang-orang(responden) untuk menjawab kuisioner tersebut.



Gambar 1.1 Kuisioner 1



Gambar 1.2 Kuisioner 2



Gambar 1.3 Kuisioner 3



Gambar 1.4 Kuisioner 4

1. Metode Wawancara

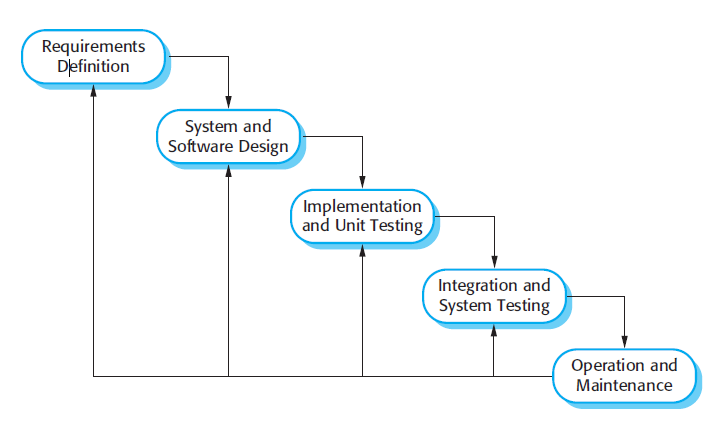
Pengumpulan data dengan cara mengadakan wawancara secara langsung kepada pengguna*(user)* yang menjadi penanggung jawab guna memperoleh data yang akurat.

1. Studi Pustaka

Dengan membaca dan meneliti dokumen-dokumen, buku-buku, referensi yang berkaitan dengan masalah yang diteliti guna mengumpulkan data dan informasi yang diperlukan.

1. Tahap pembuatan perangkat lunak

Penelitian menggunakan metode *Waterfall* dalam tahap pembuatan perangkat lunak. Pembuatan perangkat lunak dilakukan secara berurutan seperti pada gambar dibawah ini.



Gambar 1.5 Waterfall

1. **Requirements Definition**

Dalam tahapan peneliti harus mengetahui objek dari CRUD. Peneliti mengambil objek data dari formulir biodata karyawan PT. AITINDO, [Lampiran].

1. **System and Sofware Design**

Pada tahapan ini, peneliti perlu mengetahui minimal spesifikasi data dari jumlah banyaknya karyawan yang akan disimpan ke dalam database, yang kemudian dibuatlah *software* *design* aplikasi CRUD untuk melakukan proses pengolahan data. Jumlah total karyawan PT. AITINDO 53 karyawan yang setiap 1 data karyawan perlu 2 Kilobyte, total penyimpanan di dalam *database* perlu 106 Kilobyte saja. Dalam hal ini, tidak perlu ada minimum spesifikasi *hardware*, karena PC dengan spesifikasi *hardware* kecil sudah bisa menyimpan semua data tersebut.

1. **Implementation and Unit Testing**

Pada tahapan ini, peneliti melakukan proses implementasi antara data karyawan dengan software design yang telah dirancang, data karyawan ini dimabil berdasarkan Tabel 3.1. Design yang telah dibuat, diubah dalam bentuk kode-kode program.  kode-kode program yang telah dihasilkan masih pada tahap modul-modul. Diakhir tahap ini, tiap modul diuji coba tanpa diintegrasikan.

1. **Integration and System Testing**

Data dan aplikasi yang diintegrasikan dan telah diuji menjadi sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah terpenuhi.

1. **Operation and Maintenance**

Aplikasi dipasang dan digunakan pada PC kantor. Pemeliharaan termasuk perbaikan kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem sesuai kebutuhan yang baru ditemukan. Sebagai contoh adanya karyawan yang resign dan karyawan baru, ini perlu pemeliharaan aplikasi atau adanya penambahan/pengurangan data *field* dari *database*.

## 1.5 Sistematika Penulisan

Penulisan ini tersusun secara sistematis, maka masing-masing pembahasan utama dibagi dalam bab-bab yang secara garis besar adalah bagian berikut ini:

Pada bab satu, akan diuraikan mengenai Latar Belakang Masalah, Masalah, dan Batasan Masalah, Tujuan penelitian, Metode penelitian, serta Sistematika Penulisan.

Bab dua, berisi tinjauan pustaka yang didapat dari sumber-sumber relevan untuk digunakan sebagai panduan dalam penelitian serta penyusunan penulisan seperti Aplikasi, Website, Java, Netbeans, dan Database.

Bab tiga, berisi identifikasi dan analisa masalah, definisi CRUD, pembahasan masalah, struktur navigasi, rancangan storyboard, rancangan database, dan rancangan halaman antarmuka aplikasi.

Bab empat, berisi hasil implementasi dan uji coba yaitu hasil dan pengujian sistem yang berisikan rincian file hasil pembuatan aplikasi, tampilan aplikasi dan pengujian aplikasi tersebut.

Bab lima, merupakan penutup penelitian yang berisi kesimpulan dari hasil penulisan dan penelitian serta saran untuk menyempurnakan dan pengembangan dalam penulisan dan penelitian ini lebih lanjut.

# 2. TINJAUAN PUSTAKA

## 2.1 Aplikasi



Gambar 2.1 Logo Aplikasi

Menurut kamus besar bahasa Indonesia aplikasi adalah penerapan dari rancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu, [9].

Pengertian aplikasi yang lain yaitu sekumpulan elemen yang saling berinteraksi dan saling berkaitan satu dengan lainnya dalam melakukan kegiatan secara bersama untuk mencapai tujuan tertentu. Sebagai contoh aplikasi komputer yang terdiri dari *software, hardware, brainware.* Ketiga unsur tersebut saling berkaitan dan bertanggung jawab dalam *input* (masukan) dan *output* (keluaran), [10].

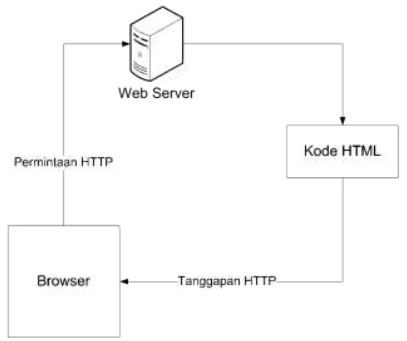
## 2.2 Website



Gambar 2.2 Logo Website

*Website* adalah kumpulan dari beberapa halaman *web* dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain dipersentasikan dalam bentuk *hypertext* dan dapat diakses oleh perangkat lunak yang disebut dengan *browser*. Informasi pada sebuah *website* pada umumnya di tulis dalam format HTML. Informasi lainya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG, dll), suara (dalam format AU, WAV, dll), dan objek multimedia lainya (seperti *MIDI, ShockwaveQuicktime Movie, 3D World, dll*), [2].

*Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hyper* *text*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* diakses dan dibaca melalui *browser* seperti *Netscape* *Navigator* atau *Internet* *Exploler* berbagai aplikasi *browser* lainnya.



Gambar 2.3 Skema Kerja HTML

### 2.2.1 Jenis-Jenis Website

1. *Website* Statis adalah suatu *website* yang mempunyai halaman yang tidak berubah. Yang artinya adalah untuk melakukan sebuah perubahan pada suatu halaman hanya bisa dilakukan secara manual yaitu dengan cara mengedit kode-kode yang menjadi struktur dari website itu sendiri.
2. *Website* Dinamis adalah merupakan suatu website yang secara strukturnya diperuntukkan untuk *update* sesering mungkin. Biasanya selain dimana utamanya yang bisa diakses oleh para pengguna (*user*) pada umumnya, juga telah disediakan halaman *backend* yaitu untuk mengedit konten dari *website* tersebut. Contoh dari *website* dinamis seperti *web* berita yang didalamnya terdapat fasilitas berita, dsb.
3. *Website* Interaktif adalah suatu *website* yang memang pada saat ini memang terkenal. Contohnya *website* interaktif seperti *forum* dan *blog*. Di *website* ini para pengguna bisa berinteraksi dan juga beradu argumen mengenai apa yang menjadi pemikiran mereka.

### 2.2.2 Fungsi-Fungsi Website

Berikut beberapa Fungsi dari *website* adalah:

1. Fungsi Komunikasi. *Website* berfungsi sebagai media komunikasi antara pembuat/pemilik dengan pengunjung atau pengunjung dengan pengunjung lain. Komunikasi dilakukan dengan menggunakan aplikasi web messanger, web forum, web chat, web mail, dan lain sebagainya.
2. Fungsi Informasi. *Website* berfungsi untuk menyediakan informasi bagi pengunjung.
3. Fungsi Hiburan. *Website* menjadi sarana hiburan, menyediakan layanan *online* *game*, *video* *streaming*, *music* *streaming*, dan lain sebagainya.
4. Fungsi Transaksi. *Website* berfungsi sebagai sarana untuk melaksanakan transaksi bisnis seperti : *online* *order*, pembayaran menggunakan kartu kredit, pembayaran dengan *e-gold*, dan sebagainya.
5. Fungsi Berbagi. *Website* sekarang ini juga dijadikan sarana berbagi berkas, seperti berkas dokumen, video, musik, dan masih banyak lagi.

### 2.2.3 Manfaat Website

Manfaat dari *website* biasanya sebagian orang memiliki suatu alasan untuk membuat web itu sendiri, diantaranya:

* Memperluas jangkauan promosi sesuatu, dengan memiliki *website* maka produk kita lebih bisa dikenal oleh masyarakat khususnya pengguna internet.
* Bisa menjadi media tanpa batas, sebab internet adalah media informasi yang tanpa batas. Dengan memiliki *website* kita berarti sama saja memiliki banayak karyawan yang mempromosikan produk kita selama 24 jam. Yang artinya dimana *website* kita akan memberikan suatu informasi kepada calon konsumen selama 24 jam.
* Promosi yang luas, internet adalah suatu media promosi terluas di dunia jika dilihat dari jangkauan area.
* Media pengenalan perusahaan, Jika kita memiliki suatu perusahaan akan lebih mudah kita mengenalkan perusahaan lewat website, karana jangkauannya internet yang luas dan pemakainya yang banyak, sehingga perusahaan kita akan dikenal oleh masyarakat banyak sehingga dapat mendatangkan calon konsumen dengan cara promosi produk lewat *website*.
* Dan lain-lain.

### 2.2.4 HTML



Gambar 2.4 Logo HTML

HTML atau *Hypertext Markup Language* adalah format data yang digunakan untuk membuat dokumen *hypertext* (teks pada komputer yang memungkinkan *user* saling mengirimkan informasi (*request-respon*)). Berikut contoh pemakaian *tag-tag* HTML sebagai berikut, [8].

<html>

<head>

<title>

---judul halaman website----

</title>

</head>

<body>

---Isi didalam web---

</body>

</html>

### 2.2.5 XML

XML merupakan suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Di mana XML adalah turunan dari SGML *(Standard Generalized Markup Language)* yang ada sebelumnya. XML *(eXtensible Markup Language)* merupakan sebuah himpunan bagian *(subset)* dari *Standard Generalized Markup Language* (SGML), yang bertujuan agar SGML secara generik dapat melayani, menerima, dan memproses dalam web dengan cara seperti yang dimungkinkan *HyperText Markup Language* (HTML) saat ini, [15].

XML didesain untuk kemudahan implementasi dan interoperabilitas dengan SGML maupun HTML. XML adalah bahasa markup yang dirancang untuk penyampaian informasi melalui *World Wide Web* (WWW) atau sering disebut web saja. HTML akan tetap terus berkembang dan akan lebih cocok dan fleksibel untuk tujuan tertentu. Pada HTML, tag biasanya melayani dua tujuan yaitu menambahkan struktur pada dokumen dan menyatakan secara tidak langsung *style* tampilan tertentu. Seperti paragraf, ukuran *heading*, *bold* dan lain-lain. Sedangkan XML merupakan bahasa yang tidak tampil dengan sendirian. XML merupakan bahasa yang nyaris secara eksklusif tidak menentukan beberapa hal yang nampak dilayar, melainkan untuk menentukan isi spesifik, tepatnya merupakan sebuah bahasa yang digunakan untuk memanipulasi pertanyaan “apa”, daripada pertanyaan “bagaimana”.

XML seperti *database* relasional, struktur dioptimasi untuk aplikasi tertentu, ini berhubungan dengan eksploitasi isi dokumen dan tentang bagaimana isi distruktur secara internal untuk menjadikan lebih bermanfaat. XML dapat memungkinkan pertukaran informasi atau data antar perangkat *(devices)* seperti: server, PCs, smart devices, aplikasi, dan situs web.Tujuan dari XML itu sendiri secara umum adalah memungkinkan SGML untuk membantu, menerima dan memproses pada program web, di mana cara ini sekarang dapat dilakukan dengan HTML. XML dibuat untuk memudahkan pelaksanaan dan *interoperability* dengan SGML dan HTML. Suatu XML menggambarkan sebuah kelas dari obyek data yang disebut documen XML yang disimpan pada komputer, dan menggambarkan sebagian tentang program yang memproses object tersebut.

### 2.2.6 Javascript



Gambar 2.5 Logo Javascript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokument HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama pada web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah perintah di sisi *user*, yang artinya di sisi browser buka di sisi *server web*, [4].

Penulisan Kode Javascript dituliskan pada file HTML. Terdapat dua cara untuk menuliskan kode-kode Java Script agar dapat ditampilkan pada halaman HTML, yaitu:

1. Javascript ditulis pada file yang sama

Untuk penulisan dengan cara ini, perintah yang digunakan adalah

<SCRIPT LANGUANGE=”JavaScript”>program javascript</SCRIPT>.

Perintah tersebut biasanya diletakkan diantara Tag <BODY>…</BODY>

Contoh Penulisan :

<HTML>

<HEAD><TITLE>……….</TITLE>

</HEAD>

<BODY>

<SCRIPT LANGUAGE=”Javascript”>

kode javascript disini

</SCRIPT>

kode HTML disini

</BODY>

</HTML>

1. Javascript ditulis pada file terpisah

Kode Javascript bisa juga kita buat dalam file terpisah dengan tujuan agar dokumen HTML isinya tidak terlalu panjang. Atribut yang digunakan adalah:

<SCRIPT SRC=”namafile.js”>…</SCRIPT>

Diantara tag <SCRIPT………> dan <SCRIPT> tidak diperlukan lagi kode Javascriptnya karena sudah dibuat dalam file taerpisah. File yang mengandung kode Javascript berekstensi .js.

### 2.2.7 Bootstrap



Gambar 2.6 Logo Bootstrap

Bootstrap adalah *framework* ataupun *tools* untuk membuat aplikasi web ataupun website yang bersifat *responsive* secara cepat, mudah dan gratis. Kata ‘*responsive’* disini berarti bahwa tampilan web (lebar dan susunan isinya) dapat berubah secara otomatis sesuai dengan lebar layar yang menampilkannya. Bootstrap terdiri dari CSS dan HTML untuk menghasilkan Grid, Layout, Tifografi, Tabel, Form, Navigasi, dan lain-lain. Di dalam Bootstrap juga sudah terdapat Javascript (jQuery Plugins) untuk menghasilkan komponen UI yang cantik seperti *Transitions, Modal, DropDown, ScrollSpy, ToolTip, PopOver, Tab,* *Alert, Button, Carousel,* dan lain-lain, [11].

Dengan *Bootstrap*, kita dapat membuat *responsive* *website* dengan cepat dan mudah dan dapat berjalan sempurna pada browser-browser modern seperti

Chrome, Firefox, Safari, Opera dan Internet Explorer.

## 2.3 Java



Gambar 2.7 Logo Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada *C* dan *C++* namun dengan sintaksis model objek yang lebih sederhana. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai Java *Virtual Machine* (*JVM*). *Java* merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun".

Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis aplikasi *desktop*, aplikasi *mobile*, dan aplikasi *web*, [4].

### 2.3.1 Kelebihan Dan Kekurangan Java

Bahasa *Java* memiliki kelebihan dan kekurangan. Kelebihan dan kekurangan tersebut adalah sebagai berikut:

**Kelebihan**

* **Berorientasi objek**

Java adalah bahasa pemrograman yang berorientasi pada objek. Java membagi program menjadi objek-objek serta memodelkan sifat dan tingkah laku masing-masing dalam menyelesaikan suatu masalah.

* **Java bersifat multiplatform**

Java dirancang untuk mendukung aplikasi yang dapat beroperasi di lingkungan jaringan berbeda. Untuk mengakomodasi hal tersebut, *Java* *compiler* membangkitkan *bytecodes* (sebuah format yang tidak tergantung pada arsitektur tertentu yang didesain untuk mengirimkan kode ke banyak platform perangkat keras dan perangkat lunak secara efisien). Java dapat dijalankan oleh banyak *platform* seperti *Linux*, *Unix*, *Windows*, *Solari*, maupun *Mac*. Java bersifat *multithread.*

* **Multithreading**

*Multithreading* adalah kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. *Thread* dalam Java memiliki kemampuan untuk memanfaatkan kelebihan multi prosessor apabila sistem operasi yang digunakan mendukung *multi* *processor*.

* **Dapat didistribusi dengan mudah**

Java memiliki *library* rutin yang lengkap untuk dirangkai pada *TCP/IP* (seperti *HTTP* dan *FTP*) dengan mudah. Kemampuan networking Java lebih kuat dan lebih mudah digunakan. Java memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah soket koneksi. Java juga mamudahkan pembuatan *CGI (Common Gateway Interface)*.

* **Bersifat dinamis**

Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang. Java bersifat dinamis dalam tahap *linking*. *Class* yang ada dapat di *link* sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di *link* dari beberapa sumber, bahkan dari sumber dalam jaringan *Internet*.

**Kekurangan**

* **Penggunaan memory yang cukup tinggi**

Bahasa pemrograman Java memang menawarkan banyak sekali fitur yang luar biasa, mulai dari kemudahan dalam menyusun *script*, hingga fitur *object* *oriented*, yang menjadi salah satu ciri khas dari bahasa pemrograman Java. Akan tetapi sayangnya, semua kelebihan tersebut harus dikompensasi dengan kebutuhan memory yang cukup besar.

Hal ini menyebabkan beberapa aplikasi Java membutuhkan *resource* *memory* yang cukup besar untuk dapat berjalan dengan baik. Begitu pula ketika seorang *developer* akan mengembangkan aplikasi Java, *developer* juga membutuhkan *memory* yang tinggi dalam mengembangkan aplikasi ini. Tentunya komputer yang di pakai harus memiliki kapasitas *RAM* terbesar untuk melancarkan *develop* agar lebih optimal.

* **Mudah didekompilasi**

Bahasa mudahnya dari istilah dekompilasi ini adalah pengambilan *source* *code*. Jadi, Java merupakan salah satu bahasa pemrograman yang mudah mengalami dekompilasi.

*Souce* *code* dan juga *script* yang dibuat dengan bahasa pemrograman *Java* dapat dengan mudah diambil dan juga dibajak oleh orang lain, sehingga memunculkan banyak *hack* dan juga peng-*copy*-an dari aplikasi yang menggunakan bahasa pemrograman *Java* tersebut.

### 2.3.2 JDK (Java Development Kit)

*JDK* adalah Perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode Java ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *JRE (Java Runtime Envirotment)*. *JDK* wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis Java, namun tidak wajib terinstall di PC yang akan menjalankan aplikasi yang dibangun dengan Java, [5].

### 2.3.3 JRE (Java Runtime Environment)

*JRE* adalah perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibangun menggunakan Java. Versi *JRE* harus sama atau lebih tinggi dengan *JDK* yang digunakan untuk membangun aplikasi Java.

### 2.3.4 IDE (Integrated Development Environment)

*IDE* adalah perangkat lunak yang digunakan untuk membangun perangkat lunak yang lain. Contohnya *Netbeans*, *Eclipse*, dll. *Netbeans IDE* dapat digunakan untuk membangun perangkat lunak berbasis *J2SE, J2EE, J2ME, JavaFX, PHP, C/C++. Ruby, Groovy* dan *Python*.

### 2.3.5 JVM (Java Virtual Machine)

*JVM* adalah inti dari Bahasa Java. *JVM* berfungsi untuk mengubah bahasa program yang kawan buat menjadi bahasa mesin, Karena bahasa Java adalah bahasa pemrograman tingkat tinggi atau mendekati bahasa manusia maka dibutuhkan pengkonversi ke bahasa mesin. *JVM* mampu menerjemahkan program yang telah dibuat ke hampir semua platform. *“write once, run everywhere”* itulah slogan dari bahasa pemrograman Java karena adanya *JVM*.

**Kesimpulannya** :

* JVM ada di dalam JRE dan JDK.
* JRE untuk “Menjalankan” program Java.
* JDK untuk “Mengkompilasi” program Java.
* NetBeans IDE untuk “Membuat” program Java.

### 2.3.6 JSP Dan Servlet

JSP *(Java Server Page)* adalah adalah halaman web dinamis berdasarkan kode-kode HTML yang disisipkan dengan kode-kode program Java, [13].

Servlet adalah adalah program Java yang berjalan di sisi server. Ada korelasi yang erat antara JSP dan servlet dimana JSP lebih berperan pada lapisan presentation, sementara servlet berperan pada lapisan logic atau middle.

## 2.4 Netbeans



Gambar 2.8 Logo Netbeans

Netbeans adalah sebuah aplikasi *Integrated* *Development* *Environment* (*IDE*) yang berbasiskan Java dari *Sun Microsystems* yang berjalan di atas *swing*. *Swing* merupakan sebuah teknologi Java untuk pengembangan aplikasi *dekstop* yang dapat berjalan pada berbagai macam *platform* seperti *windows*, *linux*, *Mac* *OS* *X* dan *Solaris*. Sebuah *IDE* merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic* *User* *Interface* *(GUI),* suatu kode *editor* atau *text*, suatu *compiler* dan suatu *debugger*, [6].

*Netbeans* juga dapat digunakan progammer untuk menulis, meng-*compile*, mencari kesalahan dan menyebarkan program *netbeans* yang ditulis dalam bahasa pemrograman Java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat professional *dekstop(J2SE)*, *enterprise* *web,* and *mobile* *applications* dengan *Java* *language*, *C/C++*, dan bahkan *dynamic* *languages* seperti *PHP*, *JavaScript*, *Groovy*, dan *Ruby*.

*NetBeans* merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah!). *Sun* *Microsystems* mendirikan proyek kode terbuka *NetBeans* pada bulan Juni 2000 dan terus menjadi sponsor utama. Dan saat ini pun *netbeans* memiliki 2 produk yaitu *Platform* *Netbeans* dan *Netbeans* *IDE*. *Platform* *Netbeans* merupakan *framework* yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi desktop dan *Platform* *NetBeans* juga menawarkan layanan-layanan yang umum bagi aplikasi *dekstop*, mengijinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi.

Fitur fitur yang terdapat dalam netbeans antara lain:

* *Smart* *Code* *Completion*: untuk mengusulkan nama variabel dari suatu tipe, melengkapi keyword dan mengusulkan tipe parameter dari sebuah method.
* *Bookmarking*: fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
* *Go* *to* *commands*: fitur yang digunakan untuk jump ke deklarasi variabel, *source* *code* atau *file* yang ada pada *project* yang sama.
* *Code* *generator*: jika kita menggunakan fitur ini kita dapat meng-*generate* *constructor*, *setter* and *getter* *method* dan yang lainnya.
* *Error* *stripe*: fitur yang akan menandai baris yang eror dengan memberi *highlight* merah.

### 2.4.1 JSE, JEE, JME Pada Netbeans

Netbeans seperti juga konsep Javasangat fleksibel. Sepanjang *library* Java tersedia, maka kita dapat melakukan pemrograman untuk jenis aplikasi apapun.

* Kita dapat membuat aplikasi *desktop* *(JSE)*
* Pemrograman *web* dan *enterprise* *(JEE)* dapat dilakukan secara *visual*
  + Koneksi *server* *database* melalui *JDBC* dapat dilakukan dari *Netbeans*, baik pada saat perancangan maupun *deployment* program.
  + Pembuatan komponen *beans.*
  + Pembuatan *Java Server Page (JSP),* *web* *module* *(service-locator dan servlet)*, *web services* dengan menggunakan *wizard* yang telah disediakan.
* Serta dapat membuat aplikasi *mobile Android* *(JME)*.

### 2.4.2 Kelebihan dan Kekurangan Netbeans

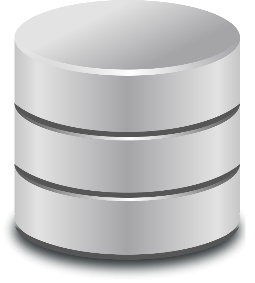
**Kelebihan**

1. *NetBeans* *GUI* *Builder* GRATIS dengan ribuan *plugin* yang bisa kita *download* langsung di *website* resminya, maupun dari pihak ketiga
2. *NetBeans* *GUI* *Builder* sangat kompetebel dengan *Swing* karena memang langsung dikembangkan oleh *Sun* yang notabenenya sebagai pengembang *Swing*.
3. *Netbeans* tidak hanya dapat digunakan buat *Java* saja, karena *Netbeans* dapat di gunakan untuk bahasa pemograman lain seperti *C/C++, Ruby,* dan *PHP*.
4. *NetBeans* *GUI* *Builder* sangat cocok untuk digunakan dalam pengembangan sistem berskala *Enterprise*.
5. Pada paket tertentu, *Netbeans* juga menyertakan *GlassFish* *V2* *UR2* dan *Apache* *Tomcat 6.0.16*.

**Kekurangan**

1. *NetBeans* hanya mendukung 1 pengembangan *Java* *GUI*, yaitu *Swing*, yang padahal ada *Java* *GUI* yang dikembangkan oleh eclipse yang bernama *SWT* dan *JFace* yang sudah cukup populer.
2. *NetBeans* mempatenkan source untuk *Java* *GUI* yang sedang dikerjakan dalam sebuah *Generated* *Code*, sehingga *programmer* tak dapat mengeditnya secara manual.
3. Dari segi sumber daya, *Netbeans* memerlukan sumber daya yang besar, seperti *Memory* dan ruang hard disk.
4. *Netbeans* memerlukan dukungan prosesor yang cukup handal untuk mendapatkan performa maksimalnya.

## 2.5 Database



Gambar 2.9 Logo Database

*Database* atau **basis data** adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data, menghidari duplikasi data, hubungan antar data yang tidak jelas dan juga update yang rumit, [7].

Proses memasukkan dan mengambil data ke dan dari *media* penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data *(database management system/DBMS)*.

### 2.5.1 DBMS

*DBMS* *(Database Management System)* merupakan sistem perangkat lunak yang memungkinkan pengguna *database* *(database user)* untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. Dengan kata lain semua akses ke basis data akan ditangani oleh *DBMS*. Ada beberapa fungsi yang harus ditangani *DBMS* yaitu mengolah pendefinisian data, menangani permintaan pemakai untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh *DBA* *(Database Administrator),* menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun disk dan menangani unjuk kerja semua fungsi secara efisien.

### 2.5.2 Macam-Macam DBMS

Macam-macam *DBMS (Database Management System)* adalah sebagai berikut:

1. **MySQL**

*MySQL* merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. Perusahaan *MySQL* *AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi *GNU* *General Public License (GPL)*, tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *GPL*. *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu *MySQL AB*. *MySQL AB* memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL* *AB* adalah: David Axmark, Allan Larsson, dan Michael Monthy Widenius.

Kelebihan *MySQL*:

* Gratis
* Stabil dan tangguh
* Fleksibel dengan berbagai pemrograman
* Security yang baik
* Dukungan dari banyak komunitas
* Kemudahan management database
* Mendukung transaksi

1. **Oracle**

*Oracle* merupakan *DBMS* yang paling rumit dan paling mahal di dunia, namun banyak orang memiliki kesan yang negatif terhadap *Oracle*. Keluhan-keluhan yang mereka lontarkan mengenai *Oracle* antara lain adalah terlalu sulit untuk digunakan, terlalu lambat dan terlalu mahal. *Oracle* adalah *relational database management system (RDBMS)* untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi. *Oracle Server* menyediakan solusi yang efisien dan efektif karena kemampuannya dalam hal sebagai berikut:

* Dapat bekerja di lingkungan *client*/*server* (pemrosesan tersebar)
* Menangani manajemen *space* dan basis data yang besar
* Mendukung akses data secara simultan
* Performansi pemrosesan transaksi yang tinggi
* Menjamin ketersediaan yang terkontrol

1. **Firebird**

*Firebird* adalah salah satu aplikasi *RDBMS* *(Relational Database Management System)* yang bersifat *open source*. Awalnya adalah perusahaan *Borland* yang sekitar tahun 2000 mengeluarkan versi *beta* dari aplikasi *database*-nya *InterBase* 6.0 dengan sifat *open* *source*. Namun entah kenapa tiba-tiba *Borland* tidak lagi mengeluarkan versi *InterBase* secara *open* *source*, justru kembali ke pola komersial *software*. Tapi pada saat yang bersamaan *programmer*-*progammer* yang tertarik dengan *source* *code* *InterBase* 6.0 tersebut lalu membuat suatu *team* untuk mengembangkan source *code* *database* ini dan kemudian akhirnya diberinama *Firebird*.

1. **Microsoft SQL Server 2000**

*Microsoft* *SQL* *Server* *2000* adalah perangkat lunak *relational database management system (RDBMS)* yang didesain untuk melakukan proses manipulasi *database* berukuran besar dengan berbagai fasilitas. *Microsoft* *SQL* *Server* 2000 merupakan produk andalan *Microsoft* untuk *database* *server*. Kemampuannya dalam manajemen data dan kemudahan dalam pengoperasiannya membuat *RDBMS* ini menjadi pilihan para *database* *administrator*.

1. **Visual Foxpro 6.0**

Pada tahun 1984, *Fox* *Software* memperkenalkan *FoxBase* untuk menyaingi *dBase II Ashton-Tate*. Pada saat itu *FoxBase* hanyalah perangkat lunak kecil yang berisi bahasa pemrograman dan mesin pengolah data. *FoxPro* memperkenalkan *GUI* *(Graphical Unit Interface)* pada tahun 1989. *FoxPro* berkembang menjadi *Visual* *FoxPro* pada tahun 1995. Kemampuan pemrogrman perihal tetap dipertahankan dan dilengkapi dengan pemrograman berorietasi objek. *Visual* *FoxPro* 6.0 dilengkapi dengan kemampuan untuk berinteraksi dengan produk *desktop* dan *client*/*server* lain dan juga dapat membangun aplikasi yang berbasis *Web*. Dengan adanya *Visual* *Studio*, *FoxPro* menjadi anggotanya. Sasaran utama *Visual* *Studio* adalah menyediakan alat bantu pemrograman dan *database* untuk mengembangka perangkat lunak yang memenuhi tuntutan zaman.

1. **Database Desktop Paradox**

*Database* *desktop* merupakan suatu program *“Add-Ins”*, yaitu program terpisah yang langsung terdapat pada *Borland* *Delphi*. Pada *database* *desktop* terdapat beberapa *DBMS* yang terintegrasi di dalamnya antara lain *Paradox* 7, *Paradox* 4, *Visual* *dBase*, *Foxpro*, *Ms* *SQL*, *Oracle*, *Ms* *Acces*, *db2* dan *interbase*. Dari beberapa *DBMS* tersebut kita akan memilih salah satu yaitu *Paradox* yang akan dibahas lebih lanjut, khususnya *Paradox* 7. Dalam *Paradox* 7 ini, pada 1 *file* *database* hanya mengizinkan 1 tabel, berbeda dengan *DBMS* lain yang mengizinkan beberapa tabel pada 1 *file* *database* seperti pada *Ms* *Acces*.

### 2.5.3 Keuntungan Dan Kekurangan DBMS

Keuntungan dari *DBMS* adalah sebagai berikut:

* Mengontrol pengulangan data.
* Data yang konsisten.
* Dapat memperoleh informasi yang lebih banyak dan jumlah data yang sama.
* Pemakaian data secara bersama-sama.
* Meningkatkan integritas data.
* Meningkatkan keamanan.
* Penetapan standarisasi.
* Perbandingan skala ekonomi.
* Menyeimbangi konflik kebutuhan.
* Meningkatkan akses dan respon dari data.
* Meningkatkan produktifitas.
* Meningkatkan pemeliharaan data melalui data independen.
* Meningkatkan konkurensi.
* Meningkatkan layanan *backup* dan *recovery*.

Kekurangan dari *DBMS* adalah sebagai berikut:

* Kompleksitas.
* Ukuran yang besar karena kompleksitas yang terjadi.
* Biaya *DBMS* yang beragam tergantung pada lingkungan dan fungsi yang disediakan.
* Tambahan biaya untuk hardware.
* Biaya konversi yang dibutuhkan untuk penggunaan *DBMS* yang baru.
* Peforma kinerja menurun.
* Tingkat kegagalan yang lebih besar.

### 2.5.4 Fasilitas DBMS

Menurut Connolly dan Begg (2010:66), fasilitas-fasilitas yang disediakan oleh *DBMS* adalah, sebagai berikut:

1. *DDL (Data Definition Language)* adalah suatu bahasa yang memperbolehkan *DBA (Database Administrator)* ataupun user untuk menentukan tipe, struktur, dan batasan keamanan data yang akan disimpan pada database.
2. *DML (Data Manipulation Language)* adalah suatu bahasa yang memiliki fasilitas untuk mengoperasikan data yang ada dalam *database*. Misalnya *insert*, *edit*, *delete*, dan *update*.
3. *SQL (Structured Query Language)* adalah suatu bahasa yang memberikan layanan akses terhadap data. Contoh dari layanan akses yang diberikan adalah:

* Sistem keamanan, yang mecegah pengguna yang tidak memiliki hak untuk mengakses *database*.
* Sistem integritas, yang mempertahankan konsistensi data yang tersimpan.
* Sistem kontrol konkurensi, yang memungkinkan untuk mengakses *database* secara bersama-sama.

### 2.5.5 RDBMS

RDBMS *(Relationship Database Management System)* adalah sebuah DBMS yang mendukung adanya hubungan *relationship* antara tabel. MySQL adalah salah satu RDBMS yang paling populer digunakan di seluruh dunia karna selain gratis MySQL juga stabil dan memiliki fitur yang lumayan lengkap.

Suatu *database* terdiri dari banyak tabel dan tabel ini terdiri dari banyak *field* yang merupakan kolomnya. Isi tiap baris dari tabel inilah merupakan data. Sedangkan untuk membuat data saling terhubung dan terintegrasi, maka satu tabel dengan tabel yang lainnya harus memiliki hubungan antara satu dengan yang lainnya. Setiap tabel mempunyai sebuah *primary key*, lalu *primary key ini*dihubungkan dengan tabel kedua dan menjadi *foreign key* di tabel kedua ini. Dengan konsep seperti inilah data akan saling terhubung dan disebut relational, karna data dalam tabel yang satu terhubung dengan tabel yang lainnya.

Berbagai macam relasi dalam database, antara lain.

* One-to-One
* One-to-Many
* Many-to-Many

### 2.5.6 JDBC

JDBC *(Java Database Connectivity)* adalah spesifikasi standar dari JavaSoft  API (*Aplication Programming Interface*) yang memungkinkan program Java untuk mengakses sistem *database* manajemen. JDBC API terdiri dari satu *set* *interface* dan kelas yang ditulis dalam bahasa pemrograman Java. JDBC API Menggunakan *interface* standar dan kelas, *programmer* dapat menulis aplikasi yang terhubung ke *database*, mengirimkan pertanyaan ditulis SQL *(Structured Query Language)*, dan memproses hasilnya, [12].

Berikut adalah empat komponen penting dari JDBC:

* 1. **JDBC-API**

JDBC-API menyediakan fasilitas untuk mengakses *database* relasional dari program berbahasa Java. Melalui komponen ini *user* dapat melakukan proses *query* dan perubahan data dalam database. JDBC-API memiliki *package* utama yang tersedia pada java.sql dan javax.sql.

* 1. **JDBC Driver Manager**

Merupakan komponen kelas utama yang mendefinisikan *object* yang mengkoneksikan aplikasi Java ke JDBC driver. Komponen ini berfungsi untuk mengatur beberapa tipe JDBC *database* *driver* yang berbeda. JDBC *Driver* Manager memilih *driver* mana yang sesuai untuk koneksi ke suatu *database*.

1. **JDBC Test Suite**

Komponen ketiga ini memastikan JDBC *driver* dapat menjalankan program user dan sangat berguna dalam melakukan testing sebuah aplikasi yang menggunakan teknologi JDBC.

1. **JDBC-ODBC Bridge**

Adalah *database driver* yang menggunakan ODBC *driver* untuk koneksi ke *database* serta memiliki fungsi utama untuk translasi dari JDBC *method* *calls* ke ODBC *function* *calls* dan mengimplementasikan JDBC untuk semua *driver* yang didukung oleh ODBC. Komponen ini dapat diimplementasikan dalam *package* sun.jdbc.odbc.

JDBC juga memiliki 10 arsitektur, yaitu :

1. Database connections
2. SQL Statements
3. Result set
4. Database metadata
5. Prepared Statement
6. Binary Large Objects (BLOBs)
7. Character Large Objects (CLOBs)
8. Callable statements
9. Database Driver
10. Driver Manager

## 2.6 Maven

Secara umum Maven disebut sebagai sebuah *build* *automation* *tool*. Maven sangat berguna bagi *programmer* Java di dalam proses pengembangan aplikasi mulai dari pembuatan *project*, mereferensi *external* *library* *(file-file JAR)*, pembuatan dokumentasi hingga compiling dan pembuatan application package (misalnya JAR atau WAR). Dengan demikian Maven adalah *tool* yang menemani *programmer* dari awal hingga akhir proses pengembangan. Maven dikembangkan oleh *Apache* *Software* *Foundation* dan tipe lisensinya adalah *free* *software* *license*. Maven ditulis dalam bahasa pemrogaman Java dan pada awal masa pembuatannya dikenal sebagai *Jakarta Project*. Informasi selengkapnya mengenai Maven dapat dijumpai pada alamat situs resmi maven:

http://maven.apache.org

Pembaca juga dapat men-download Maven secara gratis dari situs diatas, [16].

## 2.7 CRUD

Sebelum memasuki bab selanjutnya, penulis ingin memberi tahu definisi CRUD. CRUD adalah akronim dari *Create, Read, Update,* dan *Delete*. Operasi CRUD adalah manipulasi data dasar untuk *database*. Berikut penjelasan *Create, Read, Update,* dan *Delete* secara detil:

C (*Create*) adalah yang berarti membuat data baru. contoh kita sedang melakukan registrasi disebuah web itu sudah merupakan *Create* dari CRUD karena kita membuat dan menyimpan data registrasi ke *database*.

R (*Read*) adalah membaca atau menampilkan suatu data yang tadinya berada di database MySQL misalnya kemudian ditampilkan di *web* menggunakan bahasa pemrograman Java.

U (*Update*) adalah mengambil data dari *database*, yang kemudian data tersebut diperbarui menggunakan bahasa pemrograman Java.

D (*Delete*) adalah untuk melakukan penghapusan data di *database* melalui bahasa pemrograman Java.

Bagaimana cara membuat aplikasi CRUD (*create, read, update, delete*) Java Web sederhana dengan menggunakan NetBeans dan dengan koneksi database MySQL JDBC *driver*?

Dalam punulisan ini, penulis akan mencoba menjelaskan tentang pembuatan aplikasi CRUD Java Web seperti pertanyaan diatas, penjelasan sederhana mengenai aplikasi yang akan dibuat adalah dimana aplikasi nantinya akan melakukan koneksi ke *database* MySQL menggunakan koneksi JDBC *driver* dan mengeksekusi query yang diperintahkan oleh *user* untuk menampilkan data dengan perintah *Select*, memperbarui data dengan perintah SQL *Update*, menambah data dengan perintah *Insert* dan untuk menghapus data dengan perintah *Delete*.

# 3. ANALISA DAN PERANCANGAN APLIKASI

## 3.1 Analisa

Dari data yang didapat berdasarkan metode penelitian, maka dapat dibuatlah analisa yang diteliti sebagai berikut.

CRUD atau *Create, Read, Update, Delete* adalah salah satu metode pengolahan data yang sering digunakan dalam pembuatan aplikasi *desktop* maupun *web*. Secara umum, CRUD sering digunakan sebagai dasar untuk pembuatan aplikasi sederhana, dengan kata lain, CRUD juga bisa dijadikan bahan media pembelajaran dasar bahasa pemrograman. Dalam dunia IT, seorang programmer wajib bisa menguasai dasar metode ini, namun hal ini perlu sebuah objek yang perlu diimplementasikan kedalam CRUD. Sebagai media pembelajaran dasar, biodata karyawan[Lampiran] sangat tepat untuk dijadikan objek yang akan diimplementasikan kedalam aplikasi CRUD. Dalam kasus ini, peneliti menggunakan biodata karyawan PT. AITINDO sebagai objek yang akan diimplementasikan kedalam aplikasi CRUD dengan bahasa pemrograman Java berbasis *web*.

Berdasarkan uraian singkat yang telah peneliti jabarkan diatas, maka peneliti merancang dan membuat rancang aplikasi media pembelajaran CRUD dengan menggunakan bahasa pemrograman Java Web. Perancangan dalam penelitian dimulai dari perancangan struktur navigasi yang dapat dilihat pada Gambar 3.1, rancangan *database* yang dapat dilihat pada Tabel 3.1 Biodata, rancangan halaman utama yang dapat dilihat pada Gambar 3.2 Rancangan Halaman Utama, rancangan halaman tambah data yang dapat dilihat pada Gambar 3.3 Rancangan Halaman Tambah Data, dan rancangan ubah data yang dapat dilihat pada Gambar 3.4 Rancangan Halaman Ubah Data.

## 3.1 Struktur Navigasi

Struktur Navigasi digunakan sebagai penuntun alur sebuah aplikasi berbasis web atau dapat pula dianalogikan sebagai diagram alur dalam perancangan bahasa pemrograman. Struktur navigasi berfungsi untuk menggambarkan dengan jelas hubungan dan rantai kerja seluruh elemen. Adapun striktur navigasi dapat dilihat pada Gambar 3.1.

Halaman Utama

Halaman Ubah Data

Halaman Tambah Data

Gambar 3.1 Struktur Navigasi

Sebuah program membutuhkan alur atau arus yang menggambarkan program yang akan dibuat. Dari Gambar 3.1, aplikasi media pembelajaran CRUD Java Web ini memiliki 3 Halaman saja, yaitu:

1. Halaman Utama

Halaman ini merupakan halaman utama saat aplikasi dijalankan yang berfungsi untuk menampilkan semua data serta menghapus data yang dipilih dari *database*.

1. Halaman Tambah Data

Halaman ini berfungsi untuk menambah data baru, lalu akan disimpan ke *database*.

1. Halaman Ubah Data

Halaman ini berfungsi untuk memperbarui data yang diambil dari *database*.

## 3.2 Rancangan Database

Dalam perancangan aplikasi ini, peneliti menggunakan MYSQL yang telah disediakan oleh XAMPP untuk menampung data yang digunakan. Untuk melihat struktur *database* pada XAMPP yaitu dengan membuka PHPMYADMIN pada LOCALHOST di *Web Browser*. Aplikasi pembelajaran ini hanya menggunakan 1 tabel dengan nama ‘Biodata’. Struktur tabelnya adalah sebagai berikut:

Nama Database : crud.sql

Nama Tabel : Biodata

Primary Key : nik

Jumlah Field : 9

Tabel 3.1 Biodata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| 1 | Nik\* | Int | 5 | Nomor Induk Karyawan |
| 2 | Nama | Varchar | 25 | Nama karyawan |
| 3 | Jeniskelamin | Varchar | 6 | Jenis kelamin |
| 4 | Tempatlahir | Varchar | 15 | Tempat lahir |
| 5 | Tanggallahir | Varchar | 20 | Tanggal lahir |
| 6 | Alamatdomisili | Varchar | 90 | Alamat Domisili |
| 7 | Jabatan | Varchar | 25 | Jabatan |
| 8 | Agama | Varchar | 7 | Agama |
| 9 | Notelp | Varchar | 13 | No Telp/HP |

Keterangan : Tanda \* menunjukkan *Field Primary Key*

## 3.3 Rancangan Halaman Utama

Pada halaman ini berfungsi untuk menampilkan data dari *database* yang tertera di Tabel 3.1.

**APLIKASI PEMBELAJARAN CRUD JAVA WEB | NABIL FAWWAZ ELQAYYIM**

Tambah

TH10

TH7

TH6

TH5

TH1

TH9

TH8

TH3

TH4

TH2

TD7

TD6

Hapus

Ubah

TD9

TD8

TD5

TD4

TD3

TD2

TD1

Gambar 3.2 Rancangan Halaman Utama

Berikut penjelasan dari rancangan halaman utama:

1. TH1 & TD1 : *Header* Nomor dan TD1 menampilkan datanya
2. TH2 & TD2 : *Header* NIK dan TD2 menampilkan datanya
3. TH3 & TD3 : *Header* Nama dan TD3 menampilkan datanya
4. TH4 & TD4 : *Header* Jenis Kelamin dan TD4 menampilkan datanya
5. TH5 & TD5 : *Header* Tempat, Tanggal Lahir dan TD5 menampilkan datanya
6. TH6 & TD6 : *Header* Alamat Domisili dan TD6 menampilkan datanya
7. TH7 & TD7 : *Header* Jabatan dan TD7 menampilkan datanya
8. TH8 & TD8 : *Header* Agama dan TD8 menampilkan datanya
9. TH9 & TD9 : *Header* Nomor Telepon dan TD9 menampilkan datanya
10. TH10 : *Header* Aksi untuk menampilkan tombol ubah dan hapus data
11. Ubah : Tombol untuk mengubah atau memperbarui data
12. Hapus : Tombol untuk menghapus data

## 3.4 Rancangan Halaman Tambah Data

Rancangan halaman tambah data berfungsi untuk menambahkan data untuk karyawan baru.

**FORM TAMBAH DATA**

NIK

Nama

Jenis Kelamin

Pria Wanita

Tempat, Tanggal Lahir

Alamat Domisili

Jabatan

Agama

No Telp/HP

Tambah

Batal

Text1

Hasilkan NIK

Text2

Text3

Text4

Text5

Text7

Text6

Gambar 3.3 Rancangan Halaman Tambah Data

Berikut penjelasan dari rancangan halaman tambah data:

1. Text1 : untuk memasukkan data NIK
2. Text2 : untuk memasukkan data Nama
3. Text3 : untuk memasukkan data Tempat, Tanggal Lahir
4. Text4 : untuk memasukkan data Alamat
5. Text5 : untuk memasukkan data Jabatan
6. Combobox : untuk memilih data Agama
7. Text7 : untuk memasukkan data Nomor Telepon
8. Hasilkan NIK : Tombol untuk menghasilkan id Nomor Induk Karyawan
9. Tambah : Tombol untuk menyimpan semua data *field* ke *database*

## 3.5 Rancangan Halaman Ubah Data

Rancangan halaman ubah data berfungsi untuk mengubah data karyawan jika ada perubahan seperti karyawan pindah rumah, maka data alamat domisili tentu harus diperbarui.

**FORM UBAH DATA**

NIK

Nama

Jenis Kelamin

Pria Wanita

Tempat, Tanggal Lahir

Alamat Domisili

Jabatan

Agama

No Telp/HP

Perbarui

Batal

Text1

Text2

Text3

Text4

Text5

Text6

Combobox

Gambar 3.4 Rancangan Halaman Ubah Data

Berikut penjelasan dari rancangan halaman tambah data:

1. Text1 : mengambil dan mengubah data Nomor Induk Karyawan
2. Text2 : mengambil dan mengubah data Nama
3. Text3 : mengambil dan mengubah data Tempat, Tanggal Lahir
4. Text4 : mengambil dan mengubah data Alamat Domisili
5. Text5 : mengambil dan mengubah data Jabatan
6. Combobox : memilih dan mengubah data Agama
7. Text6 : mengambil dan mengubah data Nomor Telepon
8. Perbarui : Tombol untuk memperbarui data yang dipilih

# 4. HASIL IMPLEMENTASI DAN UJICOBA APLIKASI

Pada tahap ini dilakukan implementasi dari tahapan perancangan yang telah dilakukan sebelumnya, dan desain yang telah dibuat berdasarkan objek yang diambil dari Tabel 3.1 biodata. Sehingga pada tahap ini menghasilkan suatu perangkat lunak. Implementasi dilakukan untuk mengubah suatu rancangan menjadi sebuah aplikasi dengan menggunakan Java Web dan MySQL sebagai *tools*nya. Hasil implementasi diantaranya sebagai berikut:

## 4.1 Tampilan Halaman Utama

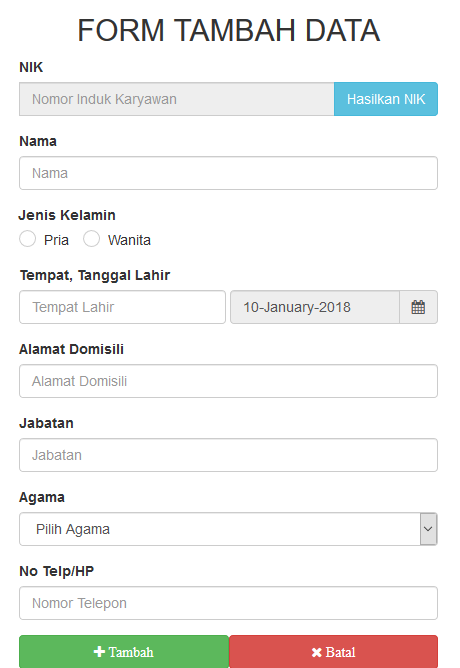
Halaman utama merupakan halaman yang pertama kali muncul sebagai tampilan awal saat aplikasi pembelajaran dijalankan melalui Netbeans. Bisa dilihat dibawah ini, halaman utama yang berdasarkan Gambar 3.2 ini berfungsi untuk menampilkan *database*, serta sebagai ruang kontrol untuk menambah, memperbarui, dan menghapus data dari *database*.



Gambar 4.1 Tampilan Halaman Utama

## 4.2 Tampilan Halaman Tambah Data

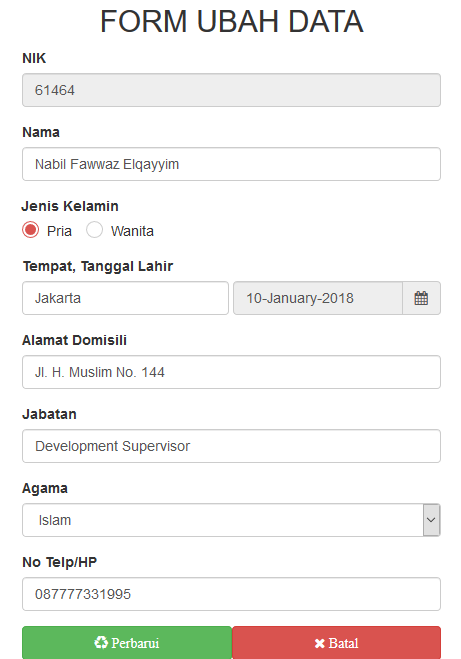
Halaman ini yang dirancang berdasarkan Gambar 3.3 ini dapat menambah data, yang nantinya akan disimpan kedalam tabel biodata pada *database* crud.sql. Untuk *field* NIK (Nomor Induk Karyawan) dan Tanggal Lahir tipenya *readonly*, tidak bisa diubah didalam *textbox*, hanya bisa diubah dengan menggunakan tombol yang ada disamping kanan *textbox.*



Gambar 4.2 Tampilan Halaman Tambah Data

## 4.3 Tampilan Halaman Ubah Data

Pada tampilan halaman ubah data yang diimplementasikan dari Gambar 3.4 ini, sebenarnya sama saja dengan tampilan halaman tambah data, yang membedakannya adalah pada tampilan halaman ubah data ini data dari *database* diambil dan dimasukkan ke dalam *field*nya masing-masing, dan tombol hijau pada halaman ini yaitu fungsinya untuk memperbarui data. Hanya *field* NIK (Nomor Induk Karyawan) saja yang bisa tidak bisa diubah.



Gambar 4.3 Tampilan Halaman Ubah Data

# 5. KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Setelah dilakukan implementasi dan uji coba aplikasi sebelumnya, Aplikasi media pembelajaran CRUD dengan bahasa pemrograman Java Web dapat bekerja dengan baik.

Maka dari pengujian tersebut dapat disimpulkan bahwa:

1. Aplikasi tersebut dapat memberikan ilmu pengetahuan tentang dasar bahasa pemrograman Java berbasis web bagi mereka yang membaca penulisan atau penelitian ini.
2. Aplikasi ini bekerja dengan sempurna baik dari sistem maupun dari design di PC kantor PT. AITINDO walaupun dengan hardware yang dibawah standar pada umumnya.

## 5.2 Saran

Rancangan aplikasi ini dapat dikembangkan lebih besar lagi, yaitu aplikasi dirancang dan dibuat dalam skala yang lebih besar dengan cara menambahkan objek data yang lebih banyak lagi seperti email, nomor rekening bank, pendidikan terakhir, dan masih banyak lagi. Serta aplikasi ditambahkan *framework* Java yang sering digunakan seperti f*ramework Spring MVC* dan *Hibernate Database*.

# DAFTAR PUSTAKA

1. Roger S, Pressman. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi* (Buku Satu). Yogyakarta: Andi.
2. Lukmanul, Hakim. 2004. *Cara Cerdas Menguasai Layout, Desain, dan Aplikasi*. Jakarta: PT Elex Media Komputindo.
3. Andry Prihantoro, Eric. 2010. *Aplikasi Web Based Learning Untuk Sekolah Menengah Atas*. Surakarta: Universitas Sebelas Maret.
4. Shalahuddin, Muhammad dan Rosa Ariani 2015. *Java Desktop Berbahasa Indonesia*. Bandung: Modula.
5. Indrajani dan Martin. 2007. *Pemrograman Berbasis Objek Dengan Bahasa Java.* Jakarta: PT Elex Media Komputindo.
6. Seliowati, Yuliana 2013. *Pendahuluan Tengan Netbeans.* Surabaya: Institut Teknologi Sepuluh November.
7. Kristanto. 2007. *Konsep & Perancangan Database*. Yogyakarta: Andi.
8. A S Rosa, Shalahudin M. 2008. *JAVA DI WEB*. Bandung: INFORMATIKA.
9. Lukmanul, Hakim. 2011. *Trik Dashyat Menguasai Ajax Dengan jQuery.* Yogyakarta: Lokomedia.
10. Jaelani, Ahmad. 2011. *Perancangan Sistem Peneriimaan Calon Siswa Baru Berbasis Web Pada SMK Putra Rifara Tangerang.* Stmik Raharja. Tanggerang.
11. Alatas, Husein. 2015. *Proyek Membangun Responsive Web Design dengan Bootstrap 3 dan 4.* Yogyakarta: Lokomedia.
12. Supardi, Yuniar. 2017. *Pemrograman Database dengan Java dan MySQL.* Jakarta: PT. Elex Media Komputindo.
13. Djuandi, Feri. 2009. *Membuat JSP dan Servlet Sederhana.* Jakarta: PT. Elex Media Komputindo.
14. Kurniawan Khamendy, Eko. 2007. *Tutorial Javascript.* Bandung: Universitas Komputer Indonesia.
15. Supriyanto, Aji. 2007. Web dengan HTML dan XML. Graha Ilmu. Yogyakarta.
16. Djuandi, Feri. 2015. *Memulai Maven.* Jakarta: PT. Elex Media Komputindo.

**LAMPIRAN**

# Formulir Biodata Karyawan

Nomor Induk Karyawan :

Nama :

Jenis Kelamin : Pria/Wanita

Tempat, Tanggal Lahir :

Alamat Domisili :

Jabatan :

Agama : Budha/Hindu/Islam/Kristen

No Telp/HP :

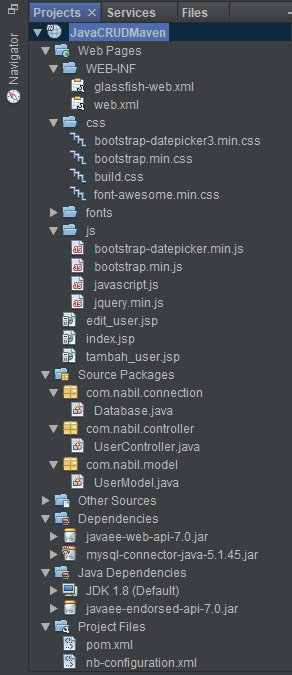
Foto

4x6

Jakarta,………..……………

( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )

## Listing Tampilan Struktur Folder Projek Aplikasi



**Download Project**

Contact: [mail@nabilfawwaz.com](mailto:mail@nabilfawwaz.com)

## Listing Coding Aplikasi

### web.xml

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">

<servlet>

<servlet-name>tambah\_user</servlet-name>

<jsp-file>/tambah\_user.jsp</jsp-file>

</servlet>

<servlet-mapping>

<servlet-name>tambah\_user</servlet-name>

<url-pattern>/tambah\_user</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>edit\_user</servlet-name>

<jsp-file>/edit\_user.jsp</jsp-file>

</servlet>

<servlet-mapping>

<servlet-name>edit\_user</servlet-name>

<url-pattern>/edit\_user</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>UserController</servlet-name>

<servlet-class>com.nabil.controller.UserController</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>UserController</servlet-name>

<url-pattern>/UserController</url-pattern>

</servlet-mapping>

<session-config>

<session-timeout>

30

</session-timeout>

</session-config>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

</web-app>

### index.jsp

<%@page import="java.util.List"%>

<%@page import="java.util.ArrayList"%>

<%@page import="com.nabil.model.UserModel"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<link rel="stylesheet" type="text/css" href="css/font-awesome.min.css">

<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">

<link rel="icon" type="image/gif" href="http://www.nabilfawwaz.com/assets/img/favicon.gif">

<title>CRUD Java Web</title>

</head>

<body>

<center><h1>Aplikasi Pembelajaran CRUD Java Web | Nabil Fawwaz Elqayyim</h1></center>

<center>

<a href="UserController?proses=input-user" class="fa fa-plus-circle btn btn-primary"> Tambah</a>

</center>

<br>

<div class=" col-lg-12">

<table class="table table-bordered table-striped table-hover" cellspacing="0" width="100%">

<thead>

<tr>

<th><i class="fa fa-list-ol"></i> No</th>

<th><i class="fa fa-vcard"></i> Nik</th>

<th><i class="fa fa-user"></i> Nama</th>

<th><i class="fa fa-venus-mars"></i> Jenis Kelamin</th>

<th><i class="fa fa-birthday-cake"></i> Tempat, Tanggal Lahir</th>

<th><i class="fa fa-address-book"></i> Alamat Domisili</th>

<th><i class="fa fa-male"></i> Jabatan</th>

<th><i class="fa fa-chain"></i> Agama</th>

<th><i class="fa fa-whatsapp"></i> Nomor Telepon</th>

<th><i class="fa fa-gears"></i> Aksi</th>

</tr>

</thead>

<%

List<UserModel> data = new ArrayList<UserModel>();

UserModel um = new UserModel();

data = um.tampil();

for (int x = 0; x < data.size(); x++) {

%>

<tr>

<td><%=x + 1%></td>

<td><%=data.get(x).getNik()%></td>

<td><%=data.get(x).getNama()%></td>

<td><%=data.get(x).getJeniskelamin()%></td>

<td><%=data.get(x).getTempatlahir() + ", " + data.get(x).getTanggallahir()%></td>

<td><%=data.get(x).getAlamatdomisili()%></td>

<td><%=data.get(x).getJabatan()%></td>

<td><%=data.get(x).getAgama()%></td>

<td><%=data.get(x).getNotelp()%></td>

<td>

<a class="btn btn-success fa fa-pencil-square-o" href="UserController?proses=edit-user&Nik=<%=data.get(x).getNik()%>"> Ubah </a>

<a class="btn btn-danger fa fa-trash-o" href="UserController?proses=hapus-user&Nik=<%=data.get(x).getNik()%>"> Hapus</a>

</td>

</tr>

<%}%>

</table>

</div>

</body>

</html>

### tambah\_user.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<link rel="stylesheet" type="text/css" href="css/font-awesome.min.css">

<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">

<link rel="stylesheet" type="text/css" href="css/bootstrap-datepicker3.min.css">

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

<link rel="icon" type="image/gif" href="http://www.nabilfawwaz.com/assets/img/favicon.gif">

<title>CRUD Java Web</title>

</head>

<body>

<center><h2>FORM TAMBAH DATA</h2></center>

<div class="col-md-4 col-md-offset-4">

<form action="UserController?data=user&proses=input-user" method="post" name="biodata">

<div class="form-group">

<div class="input-group">

<label for="NIK">NIK</label>

<input required id="nik" class="form-control" name="Nik" id="Nik" placeholder="Nomor Induk Karyawan" maxlength="5" readonly>

<span class="input-group-btn">

<button class="btn btn-info" type="button" onclick="getRandomInt()" style="margin-bottom: -25px">Hasilkan NIK</button>

</span>

</div>

</div>

<div class="form-group">

<label for="Nama">Nama</label>

<input required type="text" class="form-control" name="Nama" id="Nama" placeholder="Nama" maxlength="25" onkeypress="return onlyAlphabets(event)">

</div>

<div class="form-group">

<label class="radio" for="Jenis Kelamin" style="margin-left: -5%; margin-bottom: 3px">Jenis Kelamin</label>

<div class="radio radio-danger radio-inline">

<input type="radio" name="Jeniskelamin" id="Pria" value="Pria"><label for="Pria">Pria</label>

</div>

<div class="radio radio-danger radio-inline">

<input type="radio" name="Jeniskelamin" id="Wanita" value="Wanita"><label for="Wanita">Wanita</label>

</div>

</div>

<div class="form-group">

<label for="Tempat, Tanggal Lahir">Tempat, Tanggal Lahir</label>

<div class="form-inline">

<input required type="text" class="form-control" name="Tempatlahir" id="Tempatlahir" placeholder="Tempat Lahir" maxlength="15" style="width: 207px">

<div id="datepicker" class="input-group date" data-date-format="dd-MM-yyyy">

<input placeholder="Tanggal" class="form-control" type="text" name="Tanggallahir" id="Tanggallahir" readonly>

<span class="input-group-addon"><i class="fa fa-calendar"></i></span>

</div>

</div>

</div>

<div class="form-group">

<label for="Alamat Domisili">Alamat Domisili</label>

<input required type="text" class="form-control" name="Alamatdomisili" id="Alamatdomisili" placeholder="Alamat Domisili" maxlength="90">

</div>

<div class="form-group">

<label for="Jabatan">Jabatan</label>

<input required type="text" class="form-control" name="Jabatan" id="Jabatan" placeholder="Jabatan" maxlength="25">

</div>

<div class="form-group">

<label for="Agama">Agama</label>

<select required class="form-control" name="Agama" id="Agama">

<option value="">Pilih Agama</option>

<option value="Budha">Budha</option>

<option value="Hindu">Hindu</option>

<option value="Islam">Islam</option>

<option value="Kristen">Kristen</option>

</select>

</div>

<div class="form-group">

<label for="Nomor Telepon">Nomor Telepon</label>

<input required type="text" class="form-control" name="Notelp" id="Notelp" placeholder="Nomor Telepon" maxlength="13" onkeypress="return onlyNumeric(event)">

</div>

<button type="submit" class="btn btn-success col-md-6"><i class="fa fa-plus"> Tambah</i></button>

<a href="UserController?proses=home" class="fa fa-close btn btn-danger col-md-6"> Batal</a>

</form>

</div>

<script type="text/javascript" src="js/jquery.min.js"></script>

<script type="text/javascript" src="js/bootstrap-datepicker.min.js"></script>

<script type="text/javascript" src="js/javascript.js"></script>

<script>

$(".input-group.date").datepicker({

autoclose: true,

todayHighlight: true

}).datepicker('update', new Date());

</script>

</body>

</html>

### edit\_user.jsp

<%@page import="java.util.List"%>

<%@page import="java.util.ArrayList"%>

<%@page import="com.nabil.model.UserModel"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<link rel="stylesheet" type="text/css" href="css/font-awesome.min.css">

<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">

<link rel="stylesheet" type="text/css" href="css/bootstrap-datepicker3.min.css">

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

<link rel="icon" type="image/gif" href="http://www.nabilfawwaz.com/assets/img/favicon.gif">

<title>CRUD Java Web</title>

</head>

<body>

<div class="col-md-4 col-md-offset-4">

<center><h2>FORM UBAH DATA</h2></center>

<form action="UserController?data=user&proses=update-user" method="post" onsubmit="return input\_validation(this)">

<%

String Nik = request.getParameter("Nik");

UserModel km = new UserModel();

km.setNik(Nik);

List<UserModel> data = new ArrayList<UserModel>();

data = km.cariNik();

if (data.size() > 0) {

%>

<div class="form-group">

<label for="NIK">NIK</label>

<input required type="text" class="form-control" name="Nik" id="Nik" value="<%=data.get(0).getNik()%>" maxlength="5" readonly="readonly">

</div>

<div class="form-group">

<label for="Nama">Nama</label>

<input required type="text" class="form-control" name="Nama" id="Nama" value="<%=data.get(0).getNama()%>" maxlength="25" onkeypress="return onlyAlphabets(event)">

</div>

<div class="form-group">

<label class="radio" for="Jenis Kelamin" style="margin-left: -5%; margin-bottom: 3px">Jenis Kelamin</label>

<% if (data.get(0).getJeniskelamin().equals("Pria")) { %>

<div class="radio radio-danger radio-inline">

<input type="radio" name="Jeniskelamin" id="Pria" value="Pria" checked><label for="Pria">Pria</label>

</div>

<div class="radio radio-danger radio-inline">

<input type="radio" name="Jeniskelamin" id="Wanita" value="Wanita"><label for="Wanita">Wanita</label>

</div>

<%} else if (data.get(0).getJeniskelamin().equals("Wanita")) { %>

<div class="radio radio-danger radio-inline">

<input type="radio" name="Jeniskelamin" id="Pria" value="Pria"><label for="Pria">Pria</label>

</div>

<div class="radio radio-danger radio-inline">

<input type="radio" name="Jeniskelamin" id="Wanita" value="Wanita" checked><label for="Wanita">Wanita</label>

</div>

<%}%>

</div>

<div class="form-group">

<label for="Tempat, Tanggal Lahir">Tempat, Tanggal Lahir</label>

<div class="form-inline">

<input required type="text" class="form-control" name="Tempatlahir" id="Tempatlahir" placeholder="Tempat Lahir" maxlength="15" value="<%=data.get(0).getTempatlahir()%>" onkeypress="return onlyAlphabets(event)" style="width: 207px">

<div class="input-group date " data-date-format="dd-MM-yyyy">

<input required type="text" class="form-control" name="Tanggallahir" id="Tanggallahir" placeholder="Tanggal" value="<%=data.get(0).getTanggallahir()%>" onclick="datepicker(event)" readonly>

<span class="input-group-addon"><i class="fa fa-calendar"></i></span>

</div>

</div>

</div>

<div class="form-group">

<label for="Alamat Domisil">Alamat Domisili</label>

<input required type="text" class="form-control" name="Alamatdomisili" id="Alamatdomisili" value="<%=data.get(0).getAlamatdomisili()%>" maxlength="90">

</div>

<div class="form-group">

<label for="Jabatan">Jabatan</label>

<input required type="text" class="form-control" name="Jabatan" id="Jabatan" value="<%=data.get(0).getJabatan()%>" maxlength="25" onkeypress="return onlyAlphabets(event)">

</div>

<div class="form-group">

<label for="Agama">Agama</label>

<select required class="form-control" name="Agama" id="Agama">

<% if (data.get(0).getAgama().equals("Budha")) { %>

<option value="">Pilih Agama</option>

<option value="Budha" selected>Budha</option>

<option value="Hindu">Hindu</option>

<option value="Islam">Islam</option>

<option value="Kristen">Kristen</option>

<%} else if (data.get(0).getAgama().equals("Hindu")) { %>

<option value="">Pilih Agama</option>

<option value="Budha">Budha</option>

<option value="Hindu" selected>Hindu</option>

<option value="Islam">Islam</option>

<option value="Kristen">Kristen</option>

<%} else if (data.get(0).getAgama().equals("Islam")) { %>

<option value="">Pilih Agama</option>

<option value="Budha">Budha</option>

<option value="Hindu">Hindu</option>

<option value="Islam" selected>Islam</option>

<option value="Kristen">Kristen</option>

<%} else if (data.get(0).getAgama().equals("Kristen")) { %>

<option value="">Pilih Agama</option>

<option value="Budha">Budha</option>

<option value="Hindu">Hindu</option>

<option value="Islam">Islam</option>

<option value="Kristen" selected>Kristen</option>

<% }%>

</select>

</div>

<div class="form-group">

<label for="Nomor Telepon">Nomor Telepon</label>

<input required type="text" class="form-control" name="Notelp" id="Notelp" value="<%=data.get(0).getNotelp()%>" maxlength="13" onkeypress="return onlyNumeric(event)">

</div>

<button type="submit" class="btn btn-success col-md-6"><i class="fa fa-recycle"> Perbarui</i></button>

<a href="UserController?proses=home" class="fa fa-close btn btn-danger col-md-6"> Batal</a>

<%}%>

</form>

</div>

<script type="text/javascript" src="js/jquery.min.js"></script>

<script type="text/javascript" src="js/bootstrap-datepicker.min.js"></script>

<script type="text/javascript" src="js/javascript.js"></script>

<script>

$(".input-group.date").datepicker({

autoclose: true,

todayHighlight: true

}).datepicker('update', new Date());

</script>

</body>

</html>

### Database.java

package com.nabil.connection;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class Database {

private Connection conn;

private Statement st;

public void koneksi() {

try {

String db = "jdbc:mysql://localhost:3306/crud";

String user = "root";

String pass = "";

Class.forName("com.mysql.jdbc.Driver");

conn = DriverManager.getConnection(db, user, pass);

st = conn.createStatement();

} catch (ClassNotFoundException | SQLException x) {

System.out.println("Terjadi kesalahan koneksi : " + x);

}

}

public void diskonek(ResultSet rs) {

try {

if (rs != null) {

rs.close();

}

st.close();

conn.close();

} catch (SQLException x) {

System.out.println("Terjadi kesalahan diskoneksi : " + x);

}

}

public ResultSet ambilData(String sql) {

ResultSet rs = null;

try {

koneksi();

rs = st.executeQuery(sql);

} catch (SQLException x) {

System.out.println("Terjadi kesalahan ambil data : " + x);

}

return rs;

}

public void simpanData(String sql) {

try {

koneksi();

st.executeUpdate(sql);

} catch (SQLException x) {

System.out.println("Terjadi kesalahan simpan data : " + x);

}

}

}

### UserController.java

package com.nabil.controller;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.nabil.model.UserModel;

public class UserController extends HttpServlet {

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String proses = request.getParameter("proses");

switch (proses) {

case "input-user":

response.sendRedirect("tambah\_user");

return;

case "home":

response.sendRedirect(response.encodeRedirectURL(request.getContextPath() + "/"));

return;

case "edit-user":

response.sendRedirect("edit\_user?Nik=" + request.getParameter("Nik"));

return;

case "print":

response.sendRedirect("print?Nik=" + request.getParameter("Nik"));

return;

case "hapus-user":

UserModel hm = new UserModel();

hm.setNik(request.getParameter("Nik"));

hm.hapus();

response.sendRedirect("");

break;

default:

break;

}

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String data = request.getParameter("data");

String proses = request.getParameter("proses");

if (data != null) {

if (data.equals("user")) {

UserModel um = new UserModel();

um.setNik(request.getParameter("Nik"));

um.setNama(request.getParameter("Nama"));

um.setJeniskelamin(request.getParameter("Jeniskelamin"));

um.setTempatlahir(request.getParameter("Tempatlahir"));

um.setTanggallahir(request.getParameter("Tanggallahir"));

um.setAlamatdomisili(request.getParameter("Alamatdomisili"));

um.setJabatan(request.getParameter("Jabatan"));

um.setAgama(request.getParameter("Agama"));

um.setNotelp(request.getParameter("Notelp"));

switch (proses) {

case "input-user":

um.simpan();

break;

case "update-user":

um.update();

break;

case "hapus-user":

um.hapus();

break;

default:

break;

}

response.sendRedirect("");

}

}

}

}

### UserModel.java

package com.nabil.model;

import com.nabil.connection.Database;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class UserModel {

String id, Nik, Nama, Jeniskelamin, Tempatlahir, Tanggallahir, Alamatdomisili, Jabatan, Agama, Notelp;

Database db = null;

public UserModel() {

db = new Database();

}

public String getNik() {

return Nik;

}

public void setNik(String Nik) {

this.Nik = Nik;

}

public String getNama() {

return Nama;

}

public void setNama(String Nama) {

this.Nama = Nama;

}

public String getJeniskelamin() {

return Jeniskelamin;

}

public void setJeniskelamin(String Jeniskelamin) {

this.Jeniskelamin = Jeniskelamin;

}

public String getTempatlahir() {

return Tempatlahir;

}

public void setTempatlahir(String Tempatlahir) {

this.Tempatlahir = Tempatlahir;

}

public String getTanggallahir() {

return Tanggallahir;

}

public void setTanggallahir(String Tanggallahir) {

this.Tanggallahir = Tanggallahir;

}

public String getAlamatdomisili() {

return Alamatdomisili;

}

public void setAlamatdomisili(String Alamatdomisili) {

this.Alamatdomisili = Alamatdomisili;

}

public String getJabatan() {

return Jabatan;

}

public void setJabatan(String jabatan) {

this.Jabatan = jabatan;

}

public String getAgama() {

return Agama;

}

public void setAgama(String Agama) {

this.Agama = Agama;

}

public String getNotelp() {

return Notelp;

}

public void setNotelp(String Notelp) {

this.Notelp = Notelp;

}

public void simpan() {

String sql = "INSERT INTO biodata values('" + Nik + "','"

+ Nama + "','"

+ Jeniskelamin + "','"

+ Tempatlahir + "','"

+ Tanggallahir + "','"

+ Alamatdomisili + "','"

+ Jabatan + "','"

+ Agama + "','"

+ Notelp + "')";

db.simpanData(sql);

}

public void update() {

String sql = "UPDATE biodata SET NIK='" + Nik

+ "', NAMA='" + Nama

+ "', JENISKELAMIN='" + Jeniskelamin

+ "', TEMPATLAHIR='" + Tempatlahir

+ "', TANGGALLAHIR='" + Tanggallahir

+ "', ALAMATDOMISILI='" + Alamatdomisili

+ "', JABATAN='" + Jabatan

+ "', AGAMA='" + Agama

+ "', NOTELP='" + Notelp

+ "' WHERE Nik='" + Nik + "'";

db.simpanData(sql);

System.out.println(sql);

}

public void hapus() {

String sql = "DELETE FROM biodata WHERE NIK='" + Nik + "'";

db.simpanData(sql);

System.out.println("");

}

public List tampil() {

List<UserModel> data = new ArrayList<>();

ResultSet rs = null;

try {

String sql = "select \* from biodata order by nama asc";

rs = db.ambilData(sql);

while (rs.next()) {

UserModel um = new UserModel();

um.setNik(rs.getString("Nik"));

um.setNama(rs.getString("Nama"));

um.setJeniskelamin(rs.getString("Jeniskelamin"));

um.setTempatlahir(rs.getString("Tempatlahir"));

um.setTanggallahir(rs.getString("tanggallahir"));

um.setAlamatdomisili(rs.getString("Alamatdomisili"));

um.setJabatan(rs.getString("Jabatan"));

um.setAgama(rs.getString("Agama"));

um.setNotelp(rs.getString("Notelp"));

data.add(um);

}

db.diskonek(rs);

} catch (SQLException ex) {

System.out.println("Terjadi Kesalahan Saat menampilkan data User" + ex);

}

return data;

}

public List cariNik() {

List<UserModel> data = new ArrayList<>();

ResultSet rs = null;

try {

String sql = "SELECT \* FROM biodata WHERE Nik='" + Nik + "'";

rs = db.ambilData(sql);

while (rs.next()) {

UserModel m = new UserModel();

m.setNik(rs.getString("Nik"));

m.setNama(rs.getString("Nama"));

m.setJeniskelamin(rs.getString("Jeniskelamin"));

m.setTempatlahir(rs.getString("Tempatlahir"));

m.setTanggallahir(rs.getString("Tanggallahir"));

m.setAlamatdomisili(rs.getString("Alamatdomisili"));

m.setJabatan(rs.getString("Jabatan"));

m.setAgama(rs.getString("Agama"));

m.setNotelp(rs.getString("Notelp"));

data.add(m);

}

db.diskonek(rs);

} catch (SQLException ex) {

System.out.println("Terjadi Kesalah Saat menampilkan Cari Nik" + ex);

}

return data;

}

}