**BAB III**

**KEGIATAN PRAKTIK KERJA LAPANGAN**

Kegiatan Praktik Kerja Lapangan ini dilaksanakan dari tanggal 8 Maret 2021 sampai dengan 5 November 2021 di CV. Karya Hidup Sentosa.

1. **Produk PKL 1 : Instalasi dan Konfigurasi Software yang Diperlukan**
2. Penjelasan Singkat Jenis Pekerjaan atau Produk
3. Google Chrome



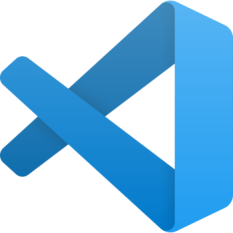
Gambar 3.1. Logo Google Chrome

Chrome adalah sebuah *web browser* yang berguna untuk menjelajah dunia maya. *Web browser* yang dikembangkan Google ini bersifat *open source* dan mempunyai nama proyek Chromium. Dengan menggunakan rendering*webkit*, maka Google Chrome mempunyai kecepatan di atas rata-rata dan tetap ringan.

Selain itu, banyak sistem operasi yang dapat digunakan Google Chrome seperti, Windows, MacOS, Linux, iOS, Android dan Chrome OS.

Google Chrome sangat cocok bagi developer web untuk mengembangkan *website*. Memiliki fitur *inspect element* yang memudahkan bagi developer web untuk menemukan *bug* atau *error* saat mengembangkan sebuah *website*. Dan memudahkan memodifikasi tampilan *website*.

1. Visual Studio Code



Gambar 3.2. Logo Visual Studio Code

Visual Studio Code merupakan aplikasi*code editorcross-platform*. *Code editor* buatan Microsoft ini memiliki fitur yang lengkap dan handal. Mendukung banyak bahasa pemrograman hingga kostumisasi tampilan yang membuat Visual Studio Code ini menjadi paket lengkap bagi para*programmer*. Ditambah banyaknya *plugins* atau *extension* membuat *code editor* ini hampir benar-benar sempurna.

1. XAMPP



Gambar 3.3. Logo XAMPP

XAMPP merupakan aplikasi *web server cross-platform* yang memudahkan bagi webmaster dan web developer dalam mengembangkan web. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi *Apahce*, *PHP*, dan *Mysql*. Saat menginstal XAMPP otomatis *Apache*, *PHP*, dan *Mysql* sudah terinstalasi dan terkonfigurasi dengan sendirinya. Sehingga memudahkan untuk membuat sebuah local *web server*.

1. DBeaver



Gambar 3.4. Logo DBeaver

Aplikasi database yang satu ini memiliki mode graphical atau GUI. Selain menyediakan versi yang berbayar, DBeaver juga menyediakan versi Community Edition yang sudah *open source* sehingga Anda bisa mengembangkannya sendiri. Ada beberapa fitur menarik yang dimiliki oleh DBeaver, yaitu didukung oleh banyak platform, dan juga memiliki kemampuan menulis berbagai file ekstensi atau *plugins*.

1. Git



Gambar 3.5. Logo Git

Git adalah salah satu sistem pengontrol versi (*Version Control System*) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds. Git akan sangat berguna bagi para developer yang kerja tim. Tugasnya mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri. Git dikenal juga dengan distributed revision control (VCS terdistribusi), dimana penyimpanan *database* Git tidak hanya di satu tempat saja.

Dengan menggunakan Git semua orang yang terlibat dalam pengkodean proyek akan menyimpan *database* Git, sehingga sangat memudahkan dalam mengelola proyek baik online maupun offline. Dengan kata lain Git sebenarnya bertugas memantau semua perubahan yang terjadi pada file proyek, lalu menyimpannya di *database*.

1. FileZilla



Gambar 3.6. Logo FileZilla

FileZilla merupakan aplikasi FTP yang *open source* dan *cross-platform*. Aplikasi ini mampu mentransfer file-file yang berukuran besar dengan aman dan cepat. Sangat cocok dan berguna untuk ruang lingkup developer. Aplikasi FileZilla mendukung FTP, SFTP, dan FTPS (FTP di SSL/TLS).

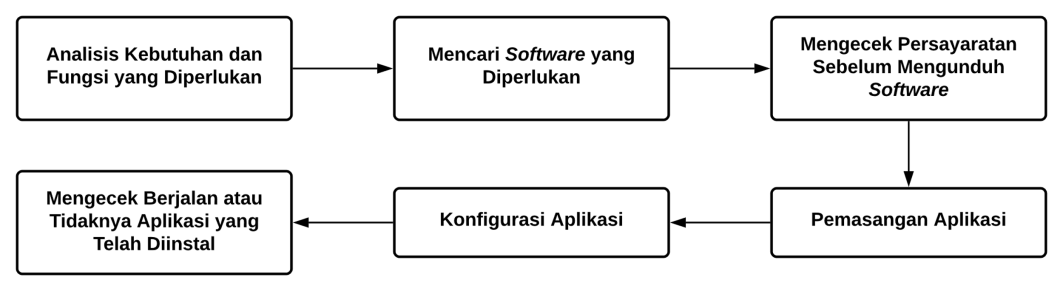
1. Codeigniter



Gambar 3.7. Logo Codeigniter

Codeigniter merupakan salah satau *framework* yang memudahkan bagi para web developer dalam membuat sebuah*website* yang berbasis *PHP*. Codeigniter menggunakan metode *MVC (Model, View, Controller)*. Dengan menggunakan codeigniter developer tidak perlu mengembangkan website dari awal lagi karena beberapa fiturnya sudah tersedia pada codeigniter. Sehingga Codeigniter akan mempersingkat waktu developer dalam mengembangkan sebuah website. Developer tinggal mengkonfigurasi dan menggunakan fungsi-fungsi yang telah disediakan oleh framework. Ditambah Codeignitermemiliki dokumentasi yang lengkap membuat codeigniter menjadi salah satu *framework* yang banyak digunakan.

1. Alat dan Bahan yang Digunakan
2. Komputer dan *peripheral*-nya
3. Sistem operasi Windows 7
4. Browser
5. Jaringan yang terkoneksi internet
6. Sumber listrik
7. Keselamatan Kerja
8. Niat dan berdoa sebelum melakukan kegiatan.
9. Memastikan semua alat dan bahan yang diperlukan sudah tersedia.
10. Menggunakan peralatan sesuai fungsinya.
11. Gunakan internet sebagaimana mestinya.
12. Posisi duduk tegap.
13. Memperhatikan jarak pandang mata dengan monitor minimal 30 cm.
14. Kurangi kecerahan monitor supaya nyaman untuk mata.
15. Hindari makan dan minum di dekat monitor.
16. Matikan perangkat bila sudah tidak digunakan.
17. Bersihkan perangkat yang digunakan bila kotor.
18. GambarKerja

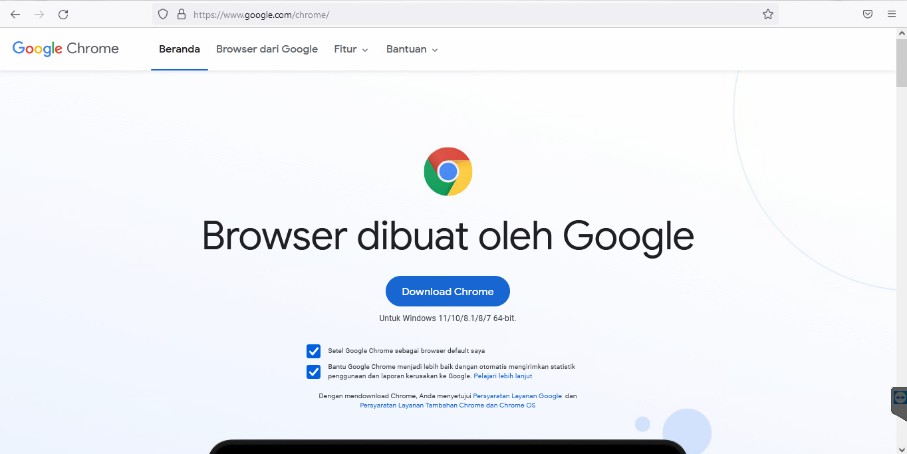
****

Gambar 3.8. Gambar Kerja Instalasi Software

1. Analisis

Instalasi dan konfigurasi software sangat diperlukan sebelum melakukan pekerjaan. Dengan instalasi dan konfigurasi software yang benar dan sesuai prosedur maka pekerjaan dapat berjalan lancar sebagaimana mestinya, kenyamanan dan etos kerja meningkat, mengurangi resiko kecelakaan kerja, meminimalisir terjadinya error, dan hasil pekerjaan menjadi lebih maksimal.

1. Langkah Kerja
2. Instalasi Google Chrome
3. Buka halaman*website* resmi Google Chrome, <https://www.google.com/chrome/>.



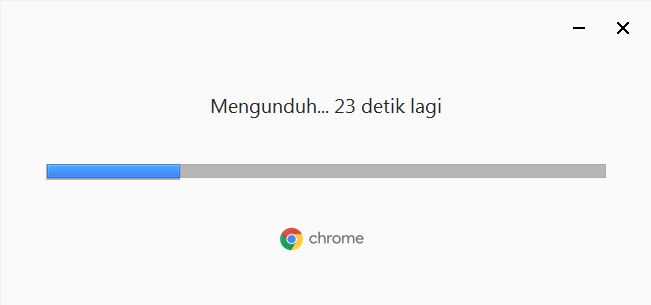
Gambar 3.9. Halaman *Website* Google Chrome

1. Klik button Download Chrome di halaman *website*, otomatis akan mengunduh *file installer* google chrome.
2. Jalankan*file installer* yang telah berhasil diunduh tadi.



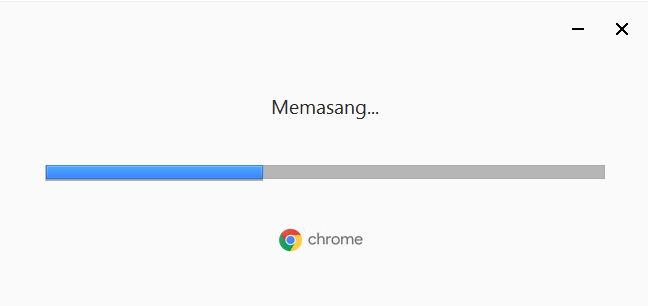
Gambar 3.10*. File Installer* Google Chrome

1. Tunggu program mengunduh aplikasi chrome.



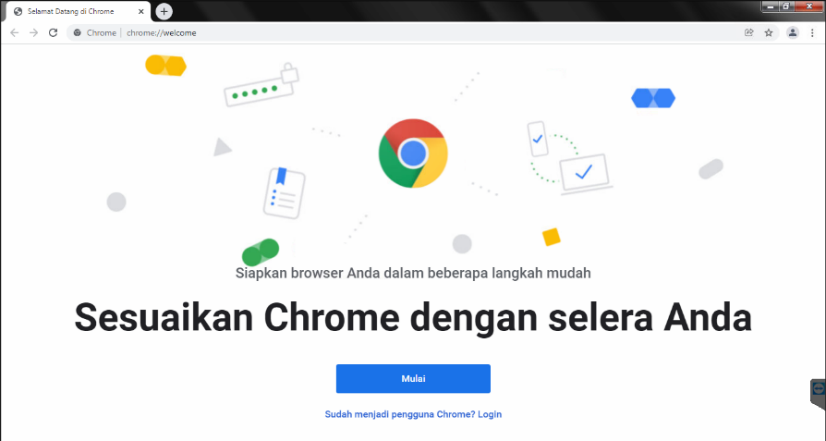
Gambar 3.11. Setup Mengunduh Google Chrome

1. Selesai mengunduh, program secara otomatis akan menginstalnya, tunggu proses instalasi.



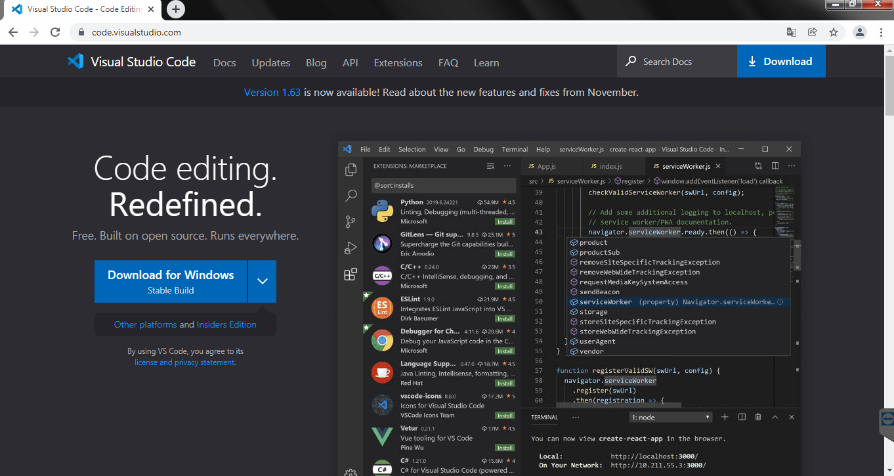
Gambar 3.12. Setup Memasang Googel Chrome

1. Jika aplikasi chrome sudah terbuka otomatis seperti ini berarti instalasi chrome telah berhasil dilakukan.*Browser* ini nantinya akan sering digunakan karena dalam pembuatan sebuah *website* memerlukan sebuah *browser*.



Gambar 3.13. Tampilan Aplikasi Google Chrome

1. Instalasi Visual Studio Code
2. Buka halaman *website* resmi Visual Studio Code, <https://code.visualstudio.com/>.



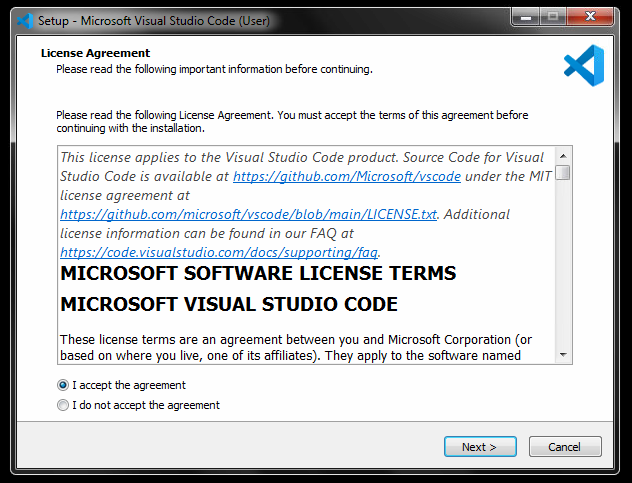
Gambar 3.14. Halaman *Website* Visual Studio Code

1. Unduh *file installer* visual studio, karena disini mengunakan windows maka pilih format installer untuk windows.
2. Jalankan file installer yang berhasil diunduh.



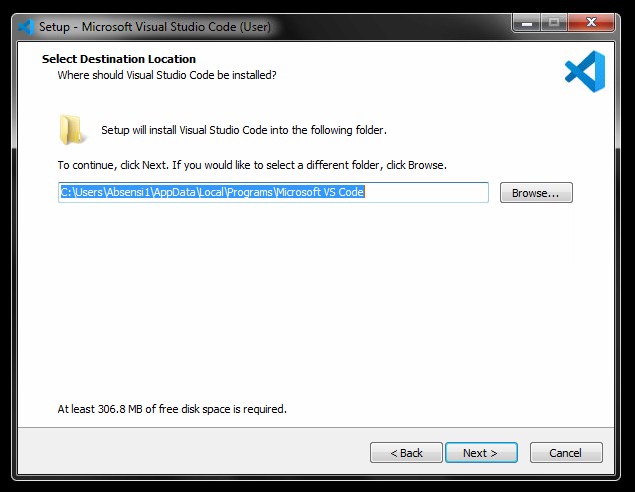
Gambar 3.15. *File Installer* Visual Studio Code

1. Pilih *I accept the agreement*, kemudian klik button next.



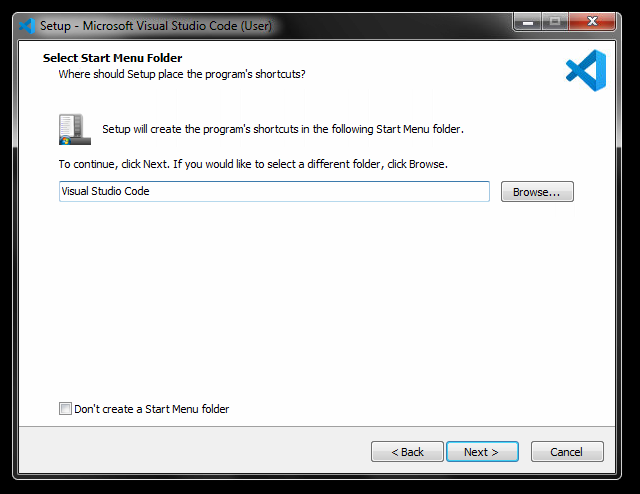
Gambar 3.16. Persetujuan Lisensi Visual Studio Code

1. Lantas pilih lokasi lokasi untuk aplikasi Visual Studio Code.



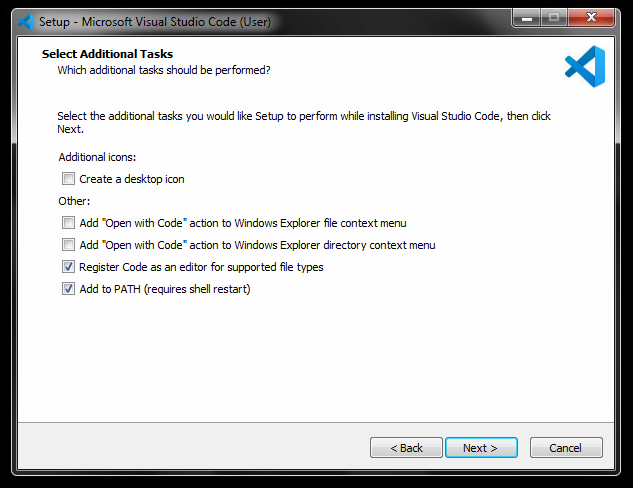
Gambar 3.17. Pemilihan Lokasi Visual Studio Code

1. Pilih lokasi untuk start menu foder, atau bisa klik next langsung jika ingin default.



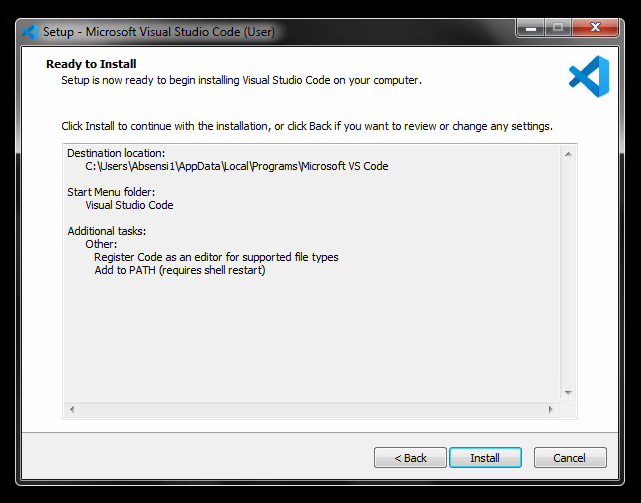
Gambar 3.18. Lokasi Start Menu Visual Studio Code

1. Kemudian pilih beberapapilihan tambahan atau bisa langsung klik next jika ingin deafult.



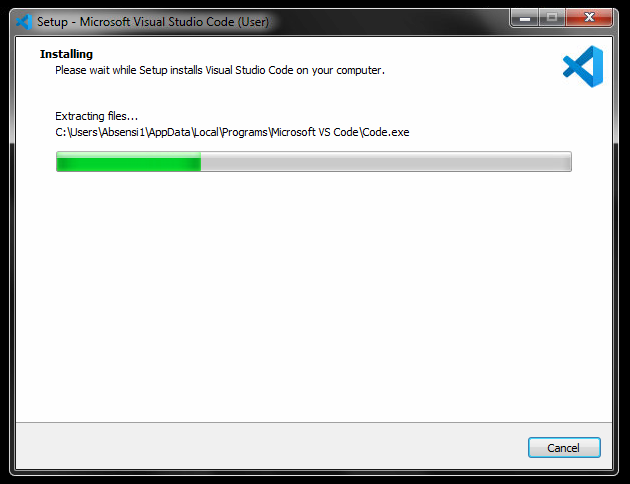
Gambar 3.19. Opsi Tambahan Instalasi Visual Studio Code

1. Maka akan muncul informasi yang sesuai dengan apa yang telah dipilih tadi, hal tersebut untuk memastikan konfigurasi. Jika sudah sesuai langsung klik tombol Install.



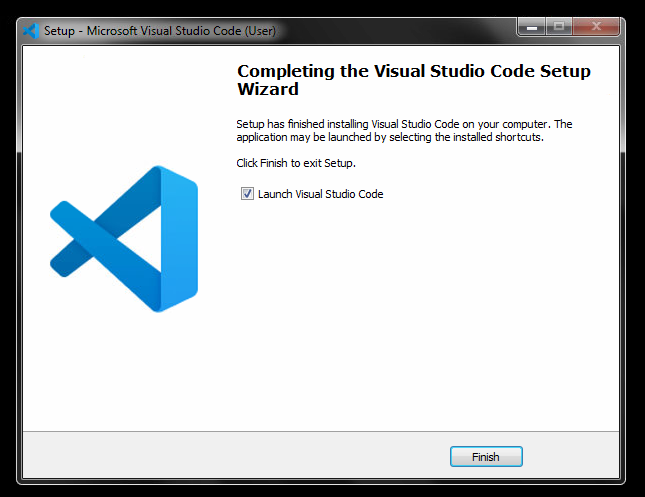
Gambar 3.20. Informasi Instalasi Visual Studio Code

1. Tunggu proses instalasi.



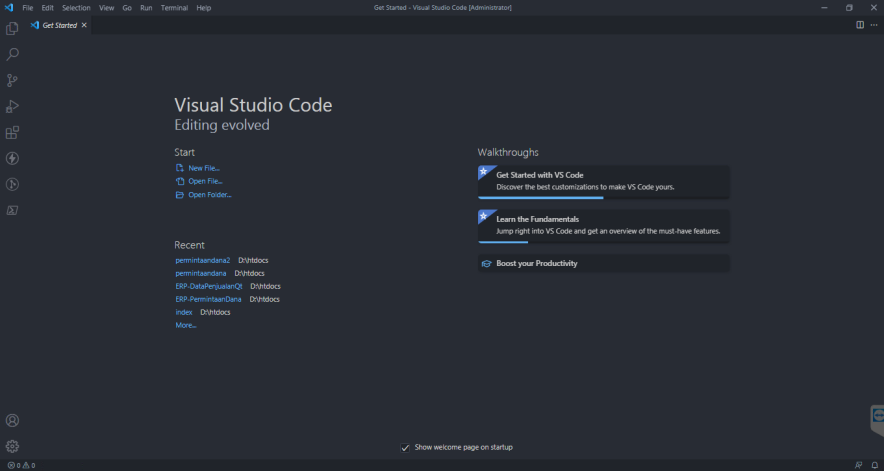
Gambar 3.21. Proses Instalasi Visual Studio Code

1. Klik tombol Finish.



Gambar 3.22. Instalasi Selesai Visual Studio Code

1. Jika Visual Studio Code sudah terbuka dan berjalan dengan normal, maka instalasi berhasil.



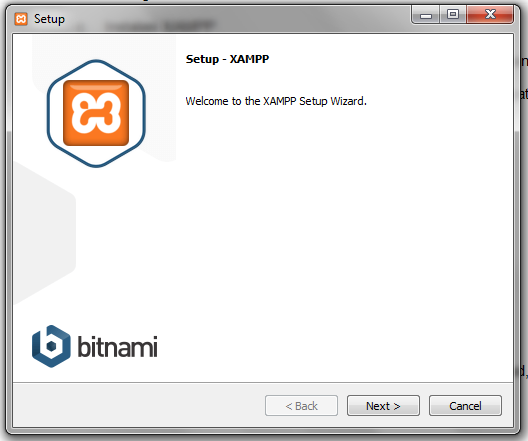
Gambar 3.23. Tampilan Aplikasi Visual Studio Code

1. Instalasi XAMPP
2. Dikarenakan PC kantor tidak dapat mengakses halaman resmi Apache untuk download XAMPP. Penulis mendapat *file installer* dari *directoryfile installer*perusahaan.



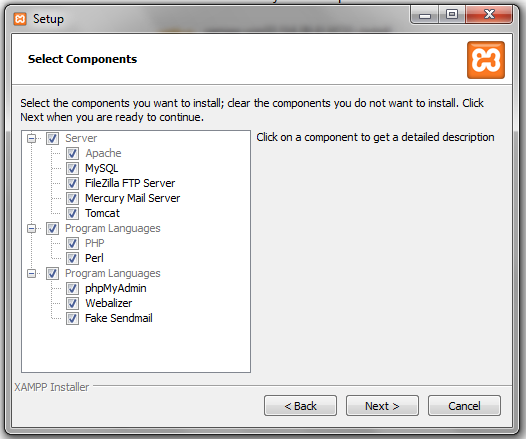
Gambar 3.24. *File Installer* XAMPP

1. Jalankan *file installer* tersebut, klik next.



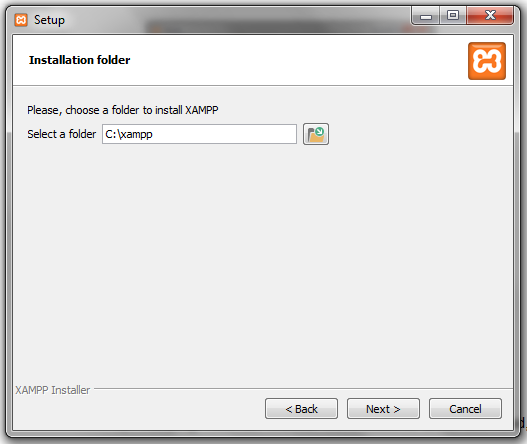
Gambar 3.25. Setup XAMPP

1. Klik next.



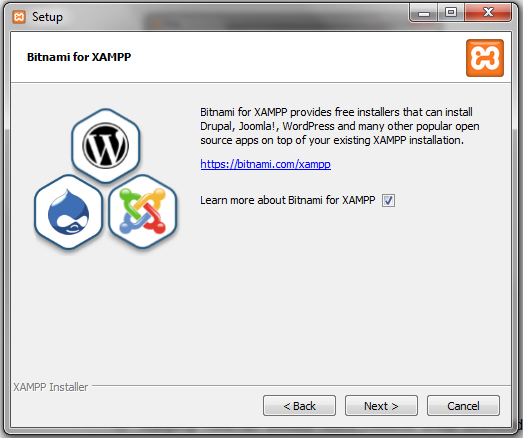
Gambar 3.26. Pilihan Komponen XAMPP

1. Pilih lokasi folder instalasi xampp, jika sudah klik next.



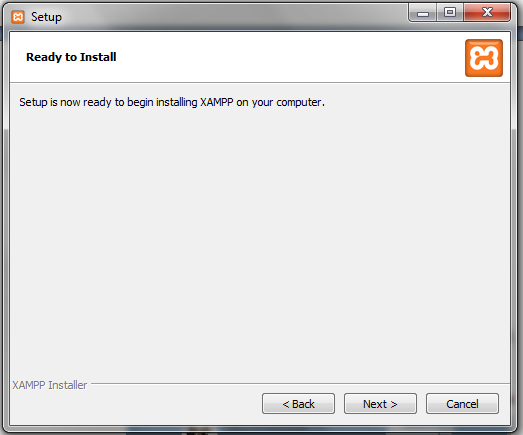
Gambar 3.27. Lokasi Folder XAMPP

1. Klik next.



Gambar 3.28. Setup XAMPP

1. Klik next.



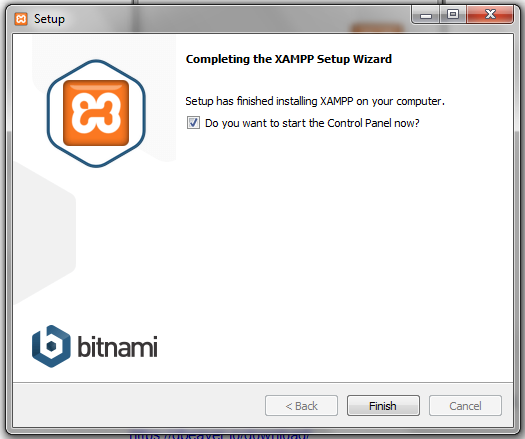
Gambar 3.29. Setup XAMPP

1. Tunggu proses instalasi hingga selesai.



Gambar 3.30. Proses Instalasi XAMPP

1. Kemudian klik finish, jika proses instalasi telah selesai.



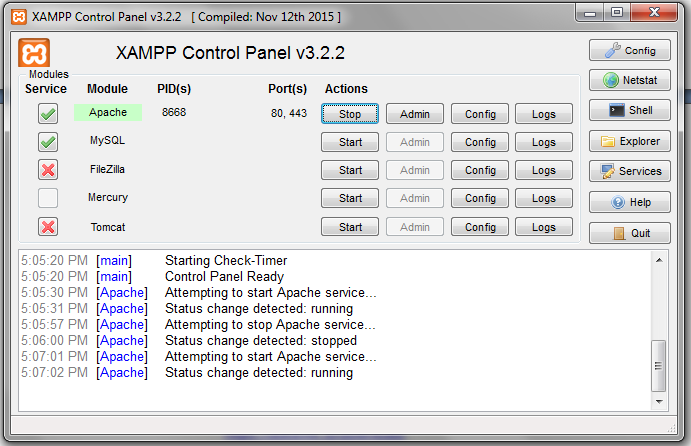
Gambar 3.31. Selesai Instalasi XAMPP

1. Pilih bahasa Inggris (bendera Amerika Serikat).



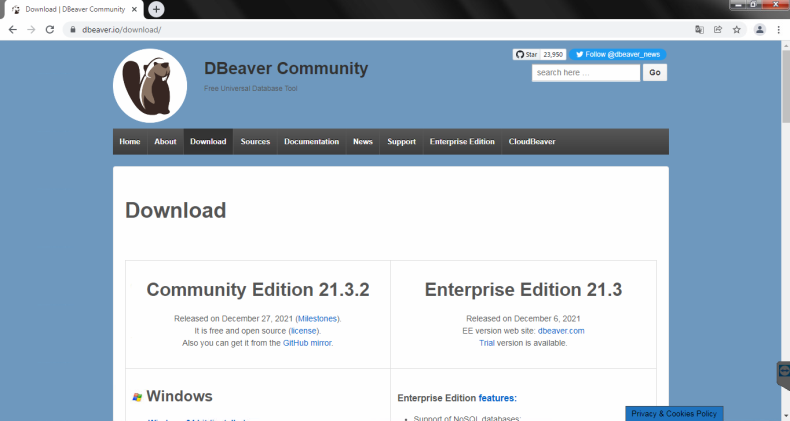
Gambar 3.32. Pemilihan Bahasa XAMPP

1. Coba jalankan service *Apache*, jika tidak ada masalah maka instalasi xampp untuk *web service* telah berhasil.



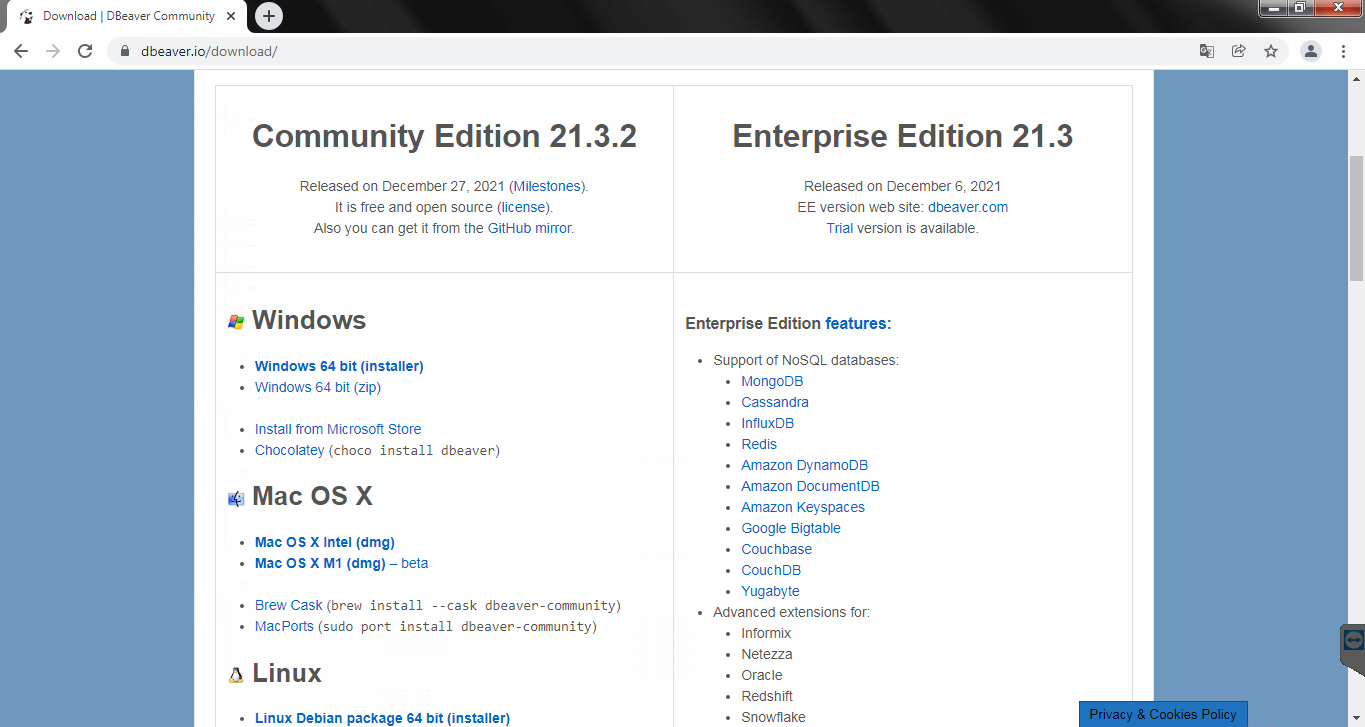
Gambar 3.33. Tampilan Aplikasi XAMPP

1. Instalasi DBevaer
2. Kunjungi halaman *website* resmi Dbeaver untuk download, <https://dbeaver.io/download/>.



Gambar 3.34. Halaman *Website* DBeaver

1. Pilih DBeaver Community Edition dan sesuaikan dengan sistem operasi yang digunakan yaitu windows.



Gambar 3.35. Pemilihan Sistem Operasi DBeaver

1. Jalankan file installer yang berhasil diunduh tadi.



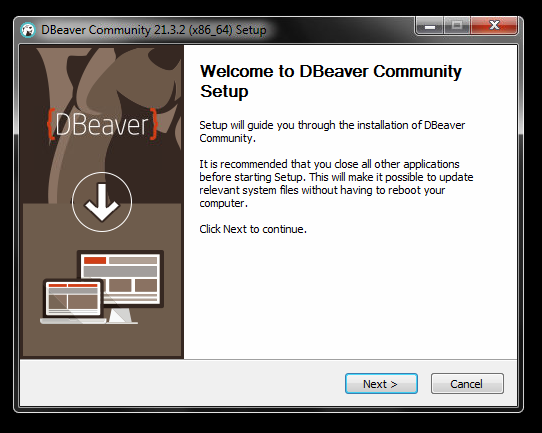
Gambar 3.36. *File Installer* DBeaver

1. Pilih bahas yang akan dipakai selama menginstall DBeaver.



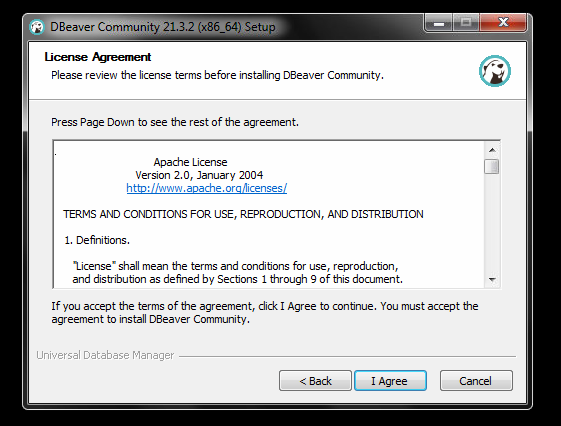
Gambar 3.37. Bahasa Instalasi DBeaver

1. Klik tombol next.



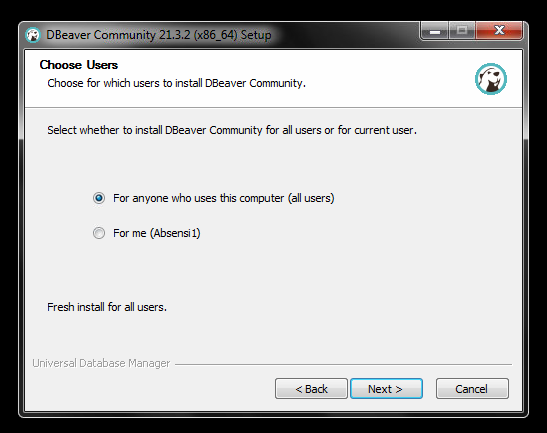
Gambar 3.38. Setup DBeaver

1. Klik tombol I Agree.



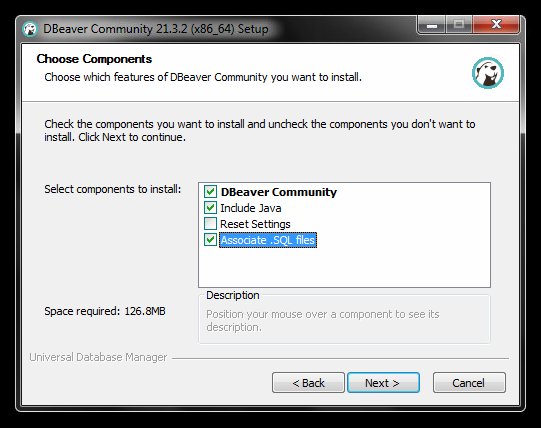
Gambar 3.39. Persetujuan Lisensi DBeaver

1. Pilih salah satu opsi dibawah, jika hendak menggunakan DBeaver untuk seluruh user komputer pilih yang atas sedangkan untuk user yang saat ini digunakan maka pilih yang bawah. Karena ini bukan komputer pribadi maka pilih yang atas.



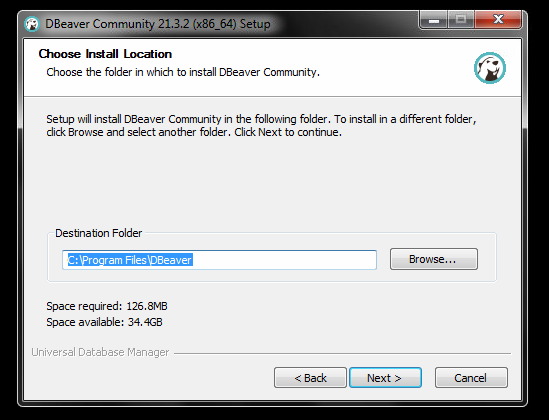
Gambar 3.40. User DBeaver

1. Pilih komponen apa saja yang akan diinstal, untuk pilihannya bisa disamakan dengan gambar.



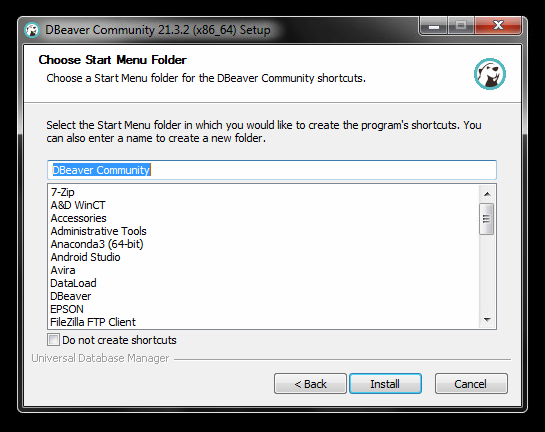
Gambar 3.41. Pilihan Komponen DBeaver

1. Pilih lokasi folder untuk DBeaver, jika ingin default langsung klik tombol next.



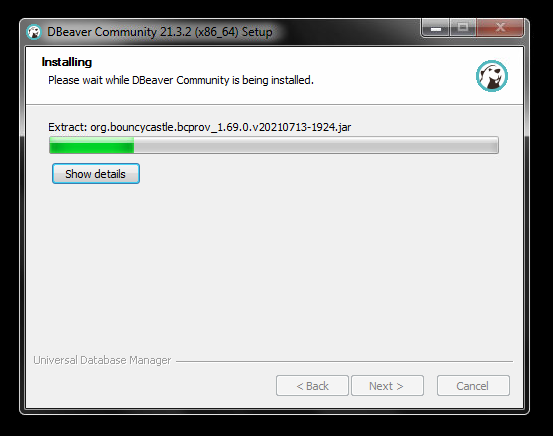
Gambar 3.42. Lokasi Folder DBeaver

1. Pilih folder start menu, jika ingin buat folder baru ubah pada kolom. Jika ingin folder default langsung klik tombol Install.



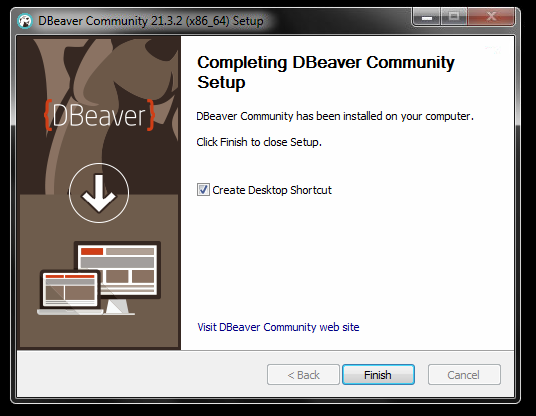
Gambar 3.43. Lokasi Folder Start Menu DBeaver

1. Tunggu proses instal DBeaver selesai.



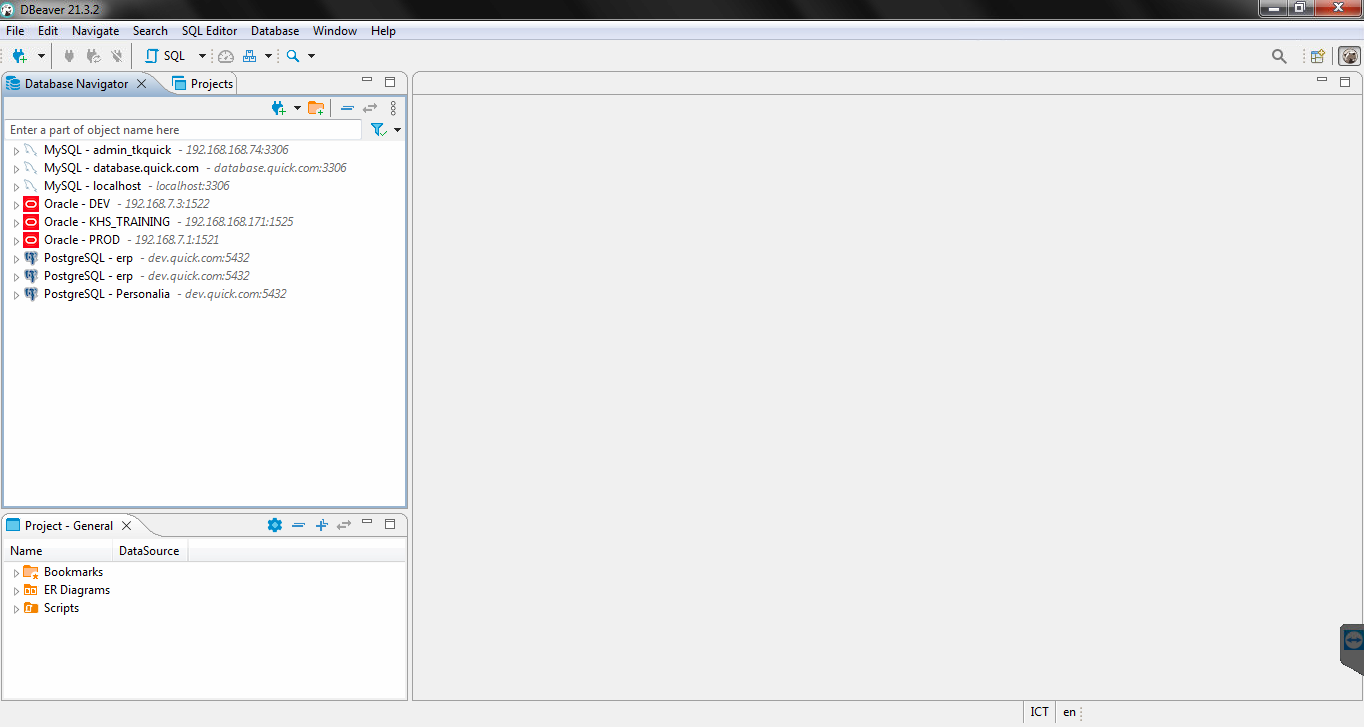
Gambar 3.44. Proses Instalasi DBeaver

1. Jika sudah selesai klik tombol finish.



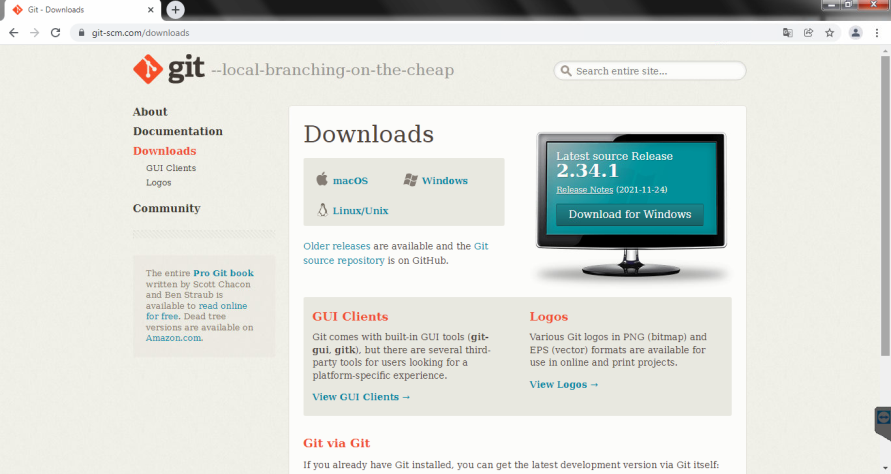
Gambar 3.45. Selesai Instalasi DBeaver

1. Kemudian jalankan aplikasi DBeaver, jika aplikasi terbuka dan normal, maka aplikasi berhasil diinstal.



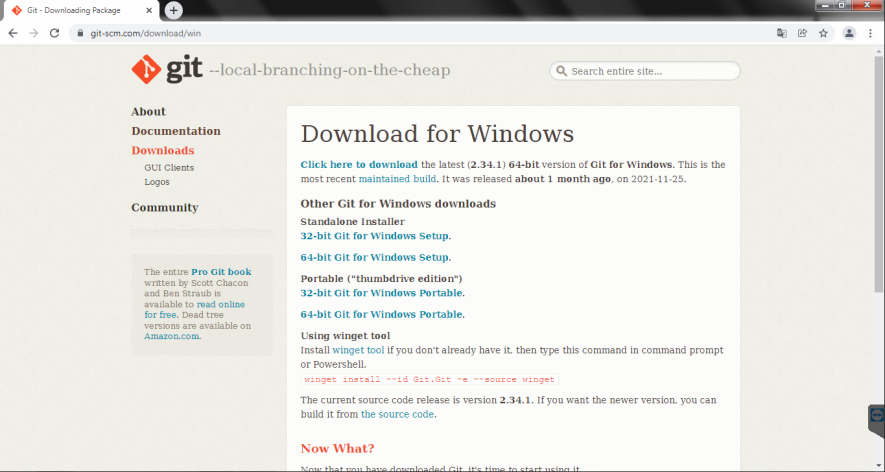
Gambar 3.46. Tampilan Aplikasi DBeaver

1. Instalasi Git
2. Buka halaman *website* resmi git untuk download, <https://git-scm.com/downloads>.



Gambar 3.47. Halaman *Website* Git

1. Pilih installer yang sesuai dengan sistem operasi, yaitu windows.
2. Kemudian pilih Standalone Installer yang 64-bit.



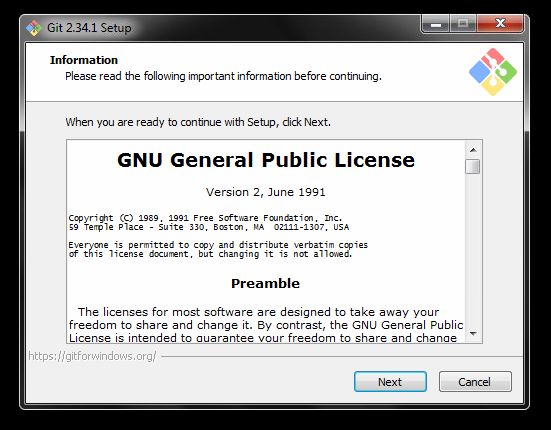
Gambar 3.48. Pemilihan Installer Git

1. Jalankan*file installer* yang berhasil diunduh tadi.



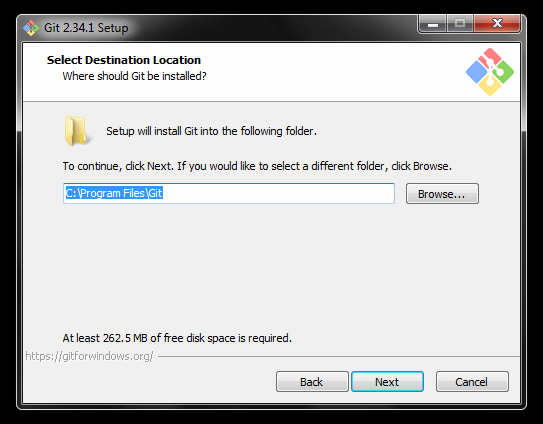
Gambar 3.49. *File Installer* Git

1. Klik tombol next.



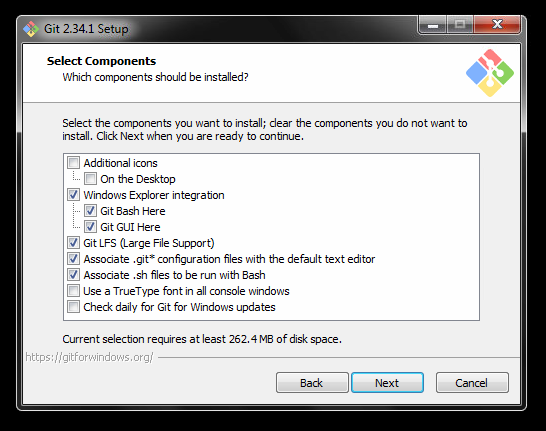
Gambar 3.50. Setup Git

1. Pilih lokasi aplikasi Git.



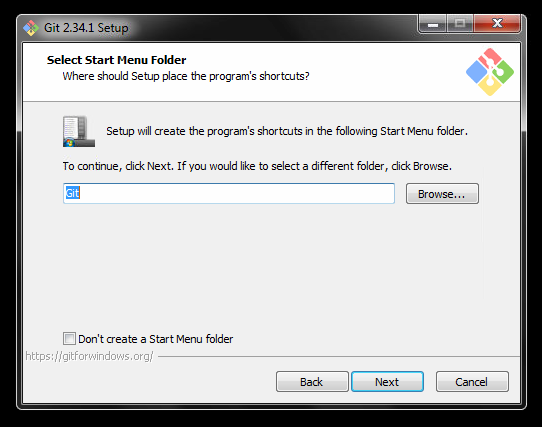
Gambar 3.51. Lokasi Folder Git

1. Pilih komponen apa saja yang akan diinstall, biarkan saja deafult langsung klik tombol next.



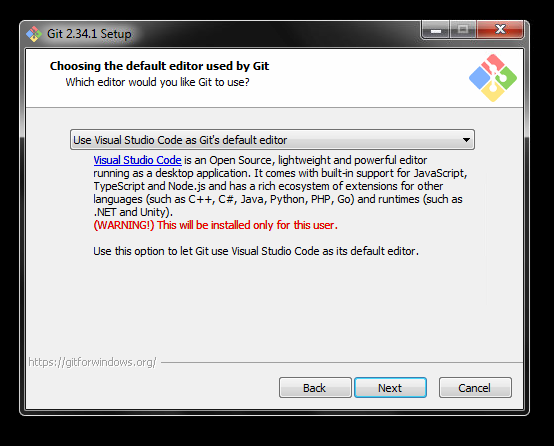
Gambar 3.52. Pilihan Komponen Git

1. Pilih lokasi folder git pada start menu, biarkan default langsung klik tombol next.



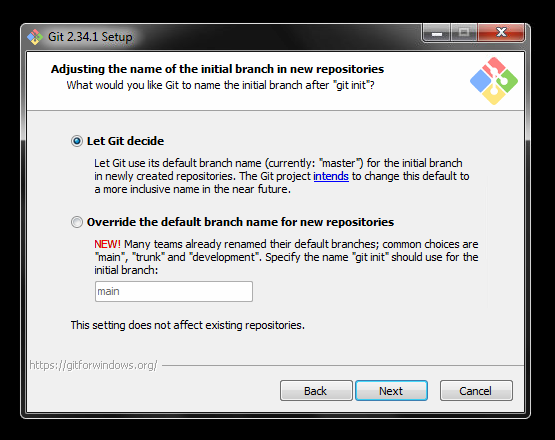
Gambar 3.53. Lokasi Folder Start Menu Git

1. Pilih *code editor* yang akan menggunakan git, karena disini menggunakan Visual Studi Code, maka pilih *Use Visual Studio Code as Git’s default editor*.



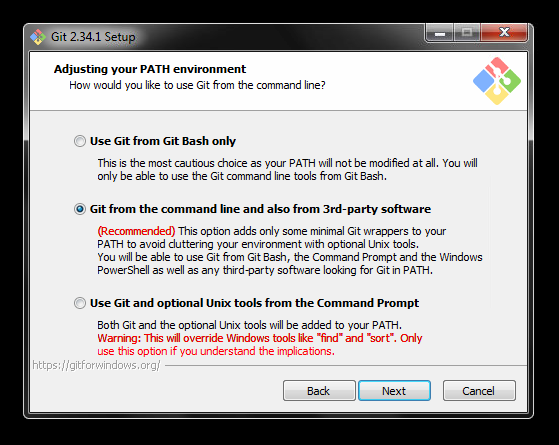
Gambar 3.54. *Code Editor* Untuk Git

1. Pilih *Let Git decide*.



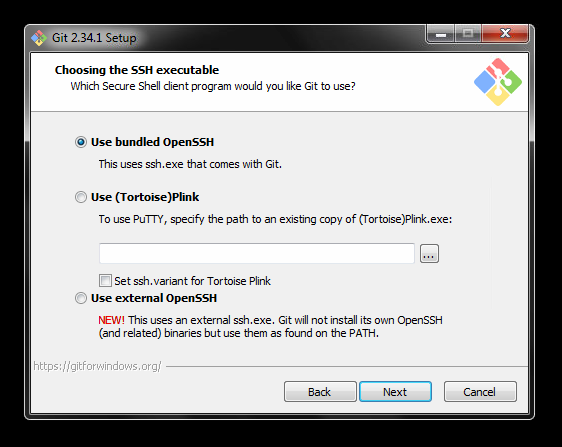
Gambar 3.55. Setup Git

1. Pilih *Git from the command line also from 3rd-party software*, klik next.



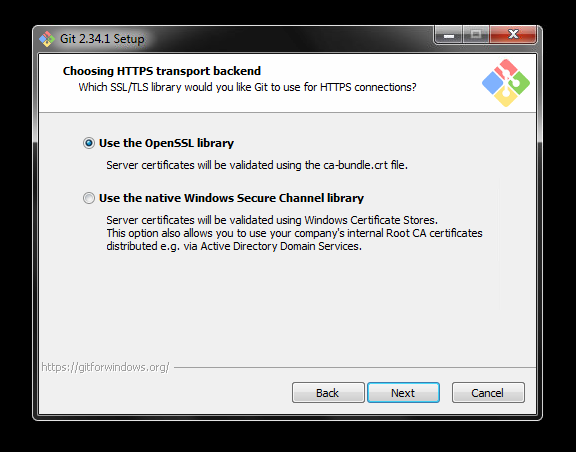
Gambar 3.56. Setup Git

1. Pilih *Use bundled OpenSSH*, klik next..



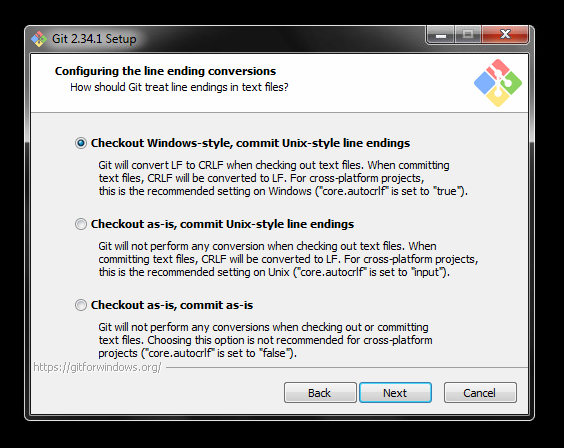
Gambar 3.57. Setup Git

1. Pilih *Use the OpenSSL library*, klik next..



Gambar 3.58. Setup Git

1. Pilih *Checkout Windows-style, commit Unix-style line endings*, klik next..



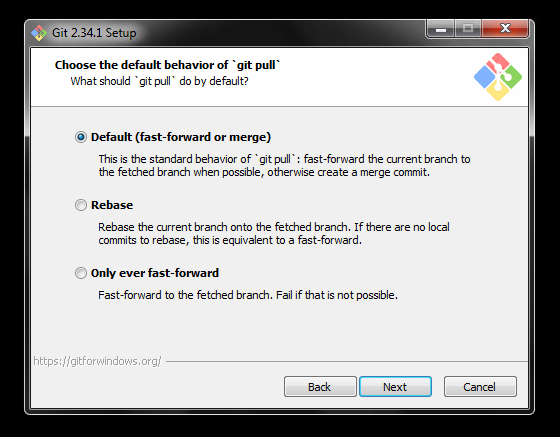
Gambar 3.59. Setup Git

1. Pilih *Use MinTTY*, klik next..



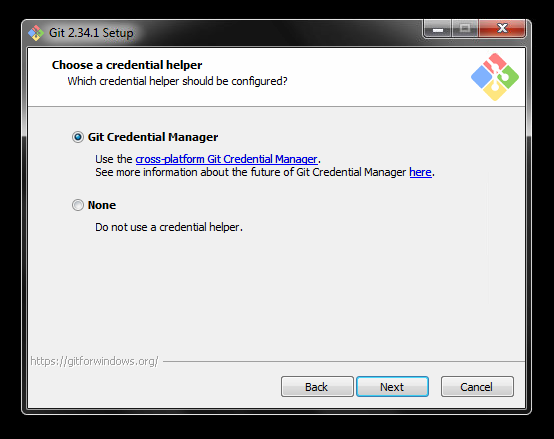
Gambar 3.60. Setup Git

1. Pilih *Default*, klik next.



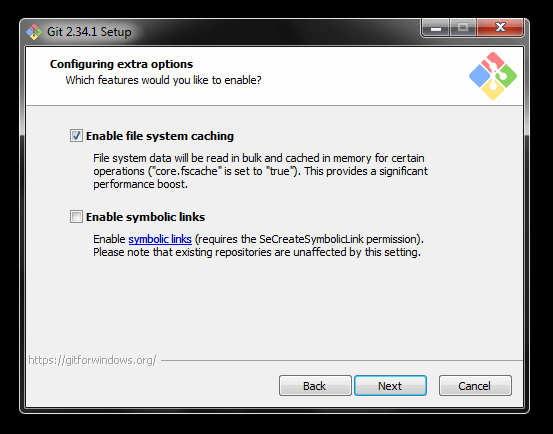
Gambar 3.61. Setup Git

1. Pilih *Git Credential Manager*, klik next.



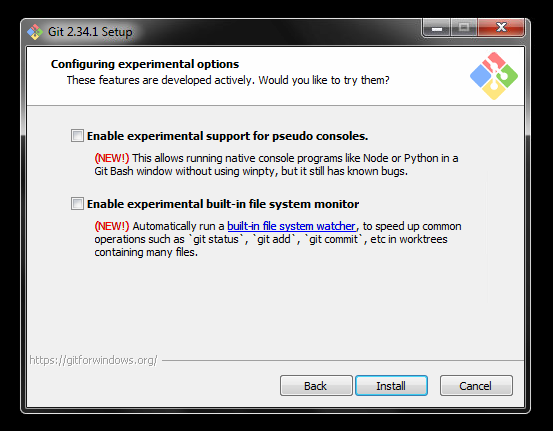
Gambar 3.62. Setup Git

1. Checklist *Enable file system caching*, klik next.



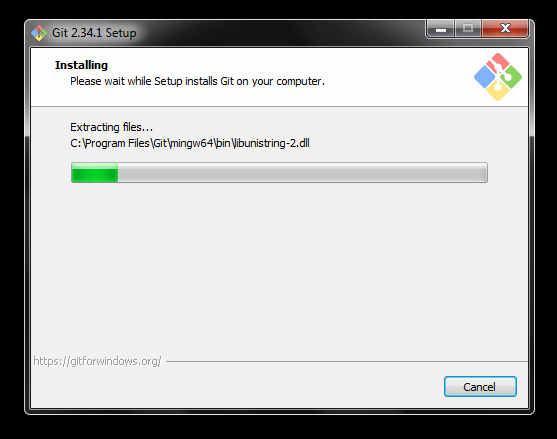
Gambar 3.63. Setup Git

1. Langsung klik tombol Install.



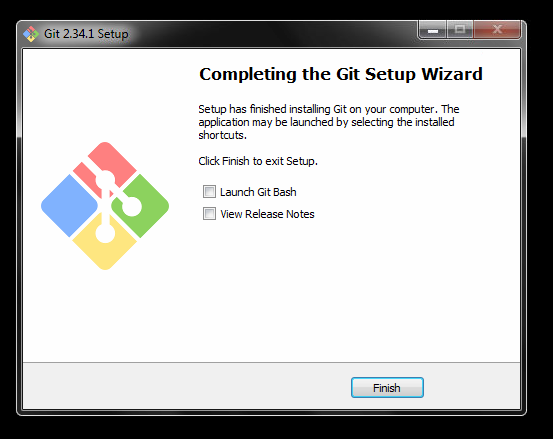
Gambar 3.64. Instal Git

1. Tunggu proses instalasi hingga selesai.



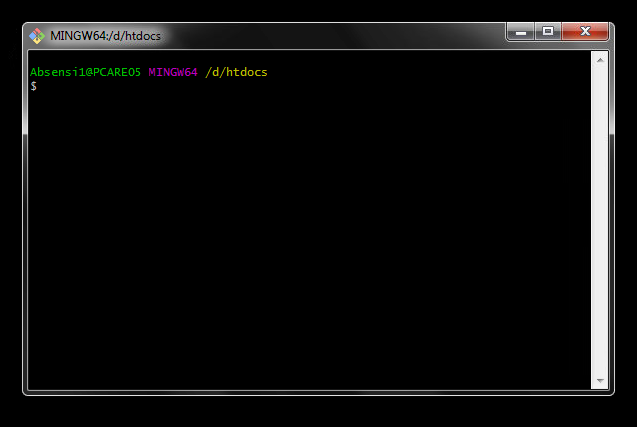
Gambar 3.65. Proses Instalasi Git

1. Jika sudah selesai klik tombol finish.



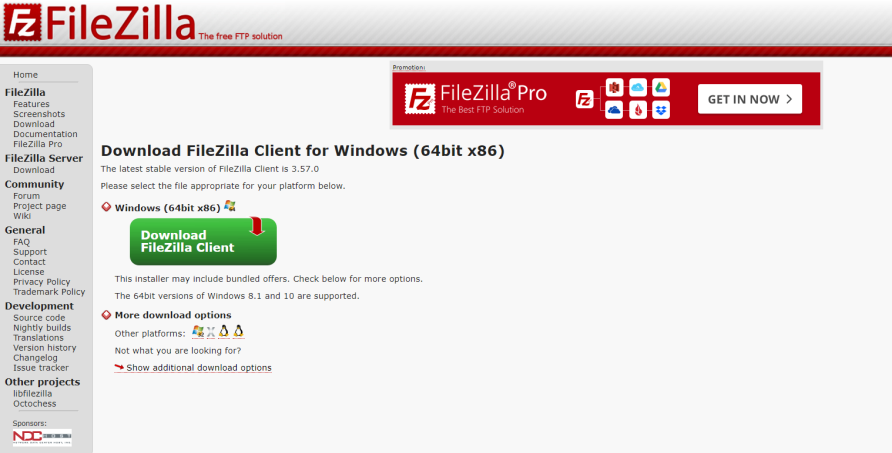
Gambar 3.66. Selesai Instalasi Git

1. Kemudian coba jalankan aplikasi gitbash, jika sudah berjalan normal seperti diubawah berarti instalasi berhasil.



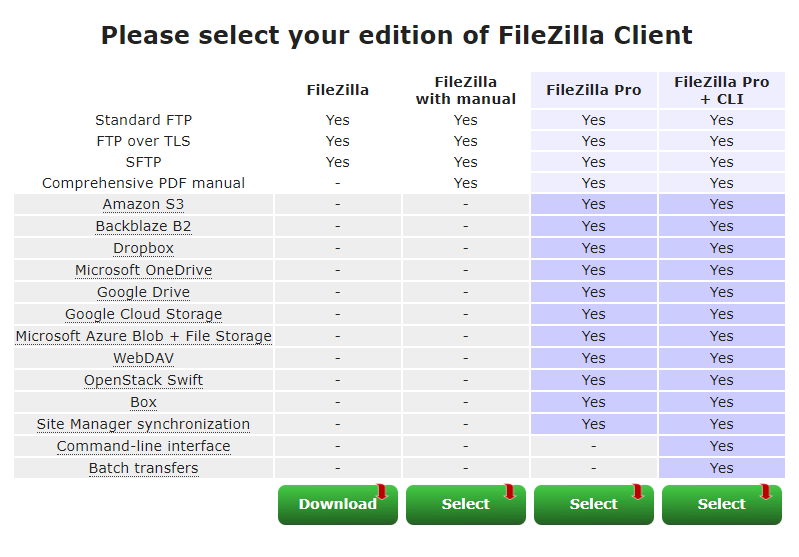
Gambar 3.67. Tampilan Aplikasi Gitbash

1. Instalasi FileZilla
2. Buka halaman*website* resmi FileZilla untuk download FileZilla Client, <https://filezilla-project.org/download.php>.



Gambar 3.68. Halaman *Website* FileZilla

1. Karena komputer disini menggunakan arsitektur 64-bit, maka tinggal klik tombol Download FileZilla Client yang berwarna hijau.
2. Kemudian pilih yang versi FileZilla yang paling kiridengan klik tombol Download dibawahnya. Versi paling kiri merupakan versi gratis.



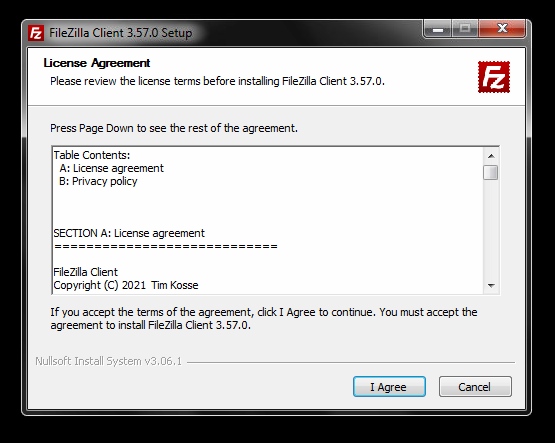
Gambar 3.69. Versi Aplikasi FileZilla

1. Jalankan *file installer* yang telah berhasil diunduh tadi.



Gambar 3.70. *File Installer* FileZilla

1. Klik tombol I Agree.



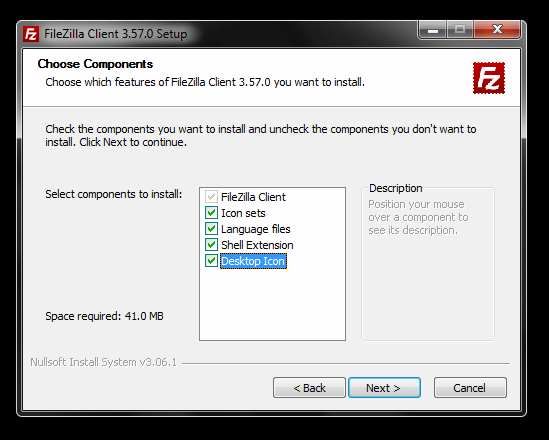
Gambar 3.71. Persetujuan Lisensi FileZilla

1. Pilih *Anyone who uses this computer*, karena ini bukan komputer pribadi.



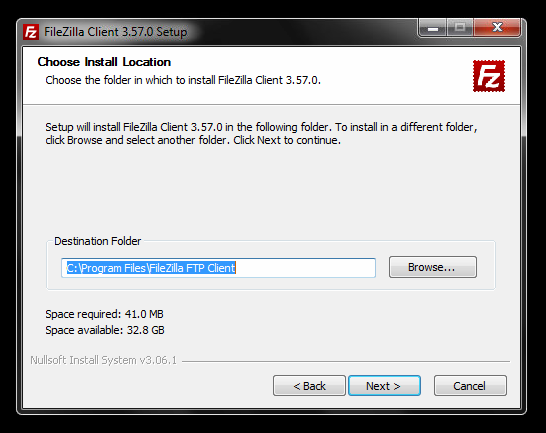
Gambar 3.72. User FileZilla

1. Untuk komponen yang akan diinstall pilih semuanya.



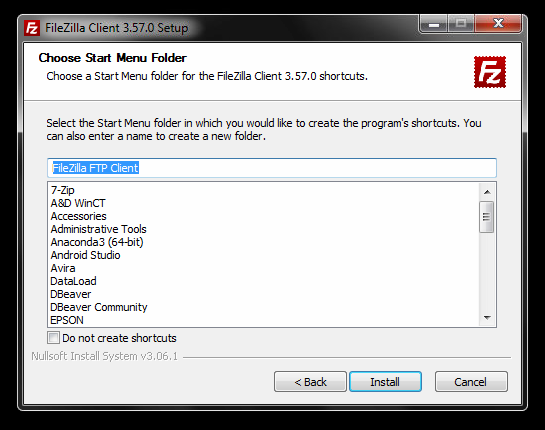
Gambar 3.73. Pilihan Komponen FileZilla

1. Untuk lokasi aplikasi FileZilla biarkan default.



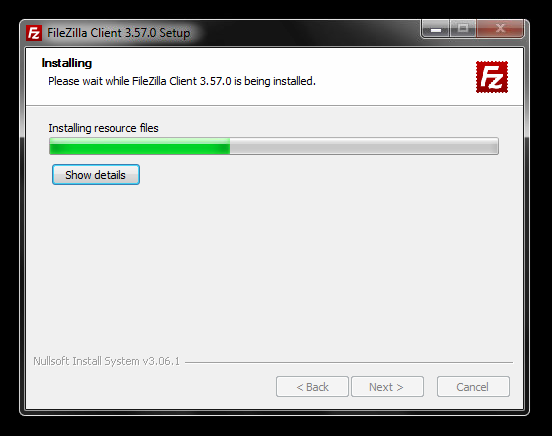
Gambar 3.74. Lokasi Folder FileZilla

1. Folder start menu untuk aplikasi FileZilla biarkan default, langsung klik tombol Install.



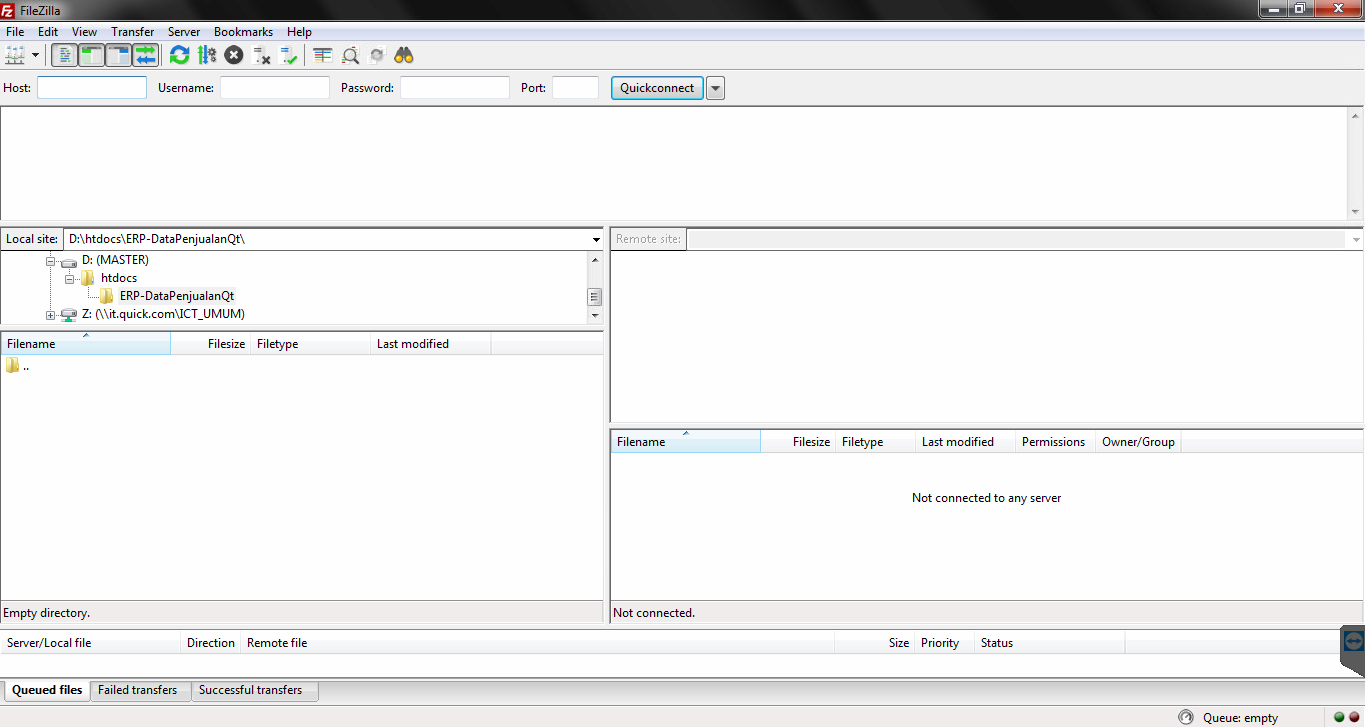
Gambar 3.75. Lokasi Folder Start Menu FileZilla

1. Tunggu proses instalasi hingga selesai.



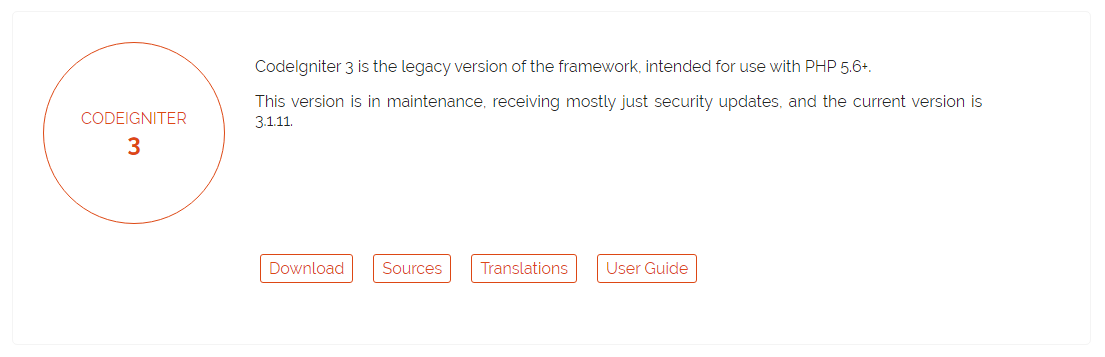
Gambar 3.76. Proses Instalasi FileZilla

1. Kemudian klik tombol Finish.
2. Jika aplikasi FileZilla sudah berjalan dengan baik, maka instalasi telah berhasil dilakukan.



Gambar 3.77. Tampilan Aplikasi FileZilla

1. Instalasi Codeigniter
2. Kunjungi halaman*webstie* resmi download Codeigniter, <https://codeigniter.com/download>.
3. Download codeigniter yang versi 3.



Gambar 3.78. Codeigniter 3

1. Hasil download berupa file dengan format zip.



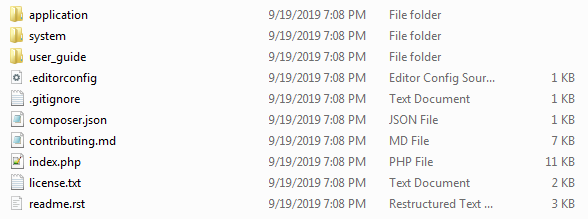
Gambar 3.79. File Setup Codeigniter 3

1. Buat sebuah folder pada *directoryweb server*.



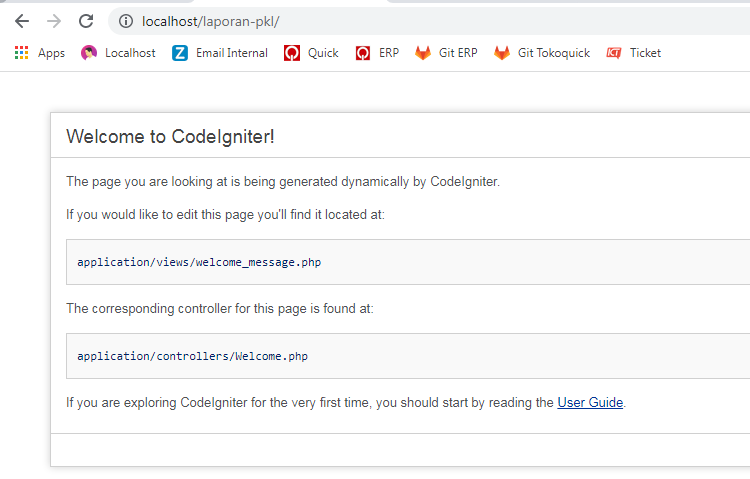
Gambar 3.80. Folder Penampung Codeigniter

1. Ekstrak file yang berhasil diunduh tadi, kemudian pindahkan file-file ini ke dalam folder yang telah dibuat tadi.



Gambar 3.81. Struktur File Codeigniter

1. Untuk memastikan berhasil atau tidaknya, buka browser dan buka alamat “localhost/nama\_folder”.



Gambar 3.82. Tampilan *Website* Codeigniter

1. Pelaksanaan Quality Control

Setelah aplikasi yang diperlukan selesai terpasang pada perangkat, langkah selanjutnya adalah menguji compatibility aplikasi pada perangkat. Pengujian compatibility pada perangkat dilakukan dengan memulai dari menjalankan fungsi dasar dan fungsi utama yang paling diperlukan.

1. Kesimpulan

Instalasi dan Konfigurasi aplikasi sangatlah penting dilakukan sebelum melakukan pekerjaan. Karena dengan menginstal aplikasi kita baru bisa melakukan pekerjaan selanjutnya.

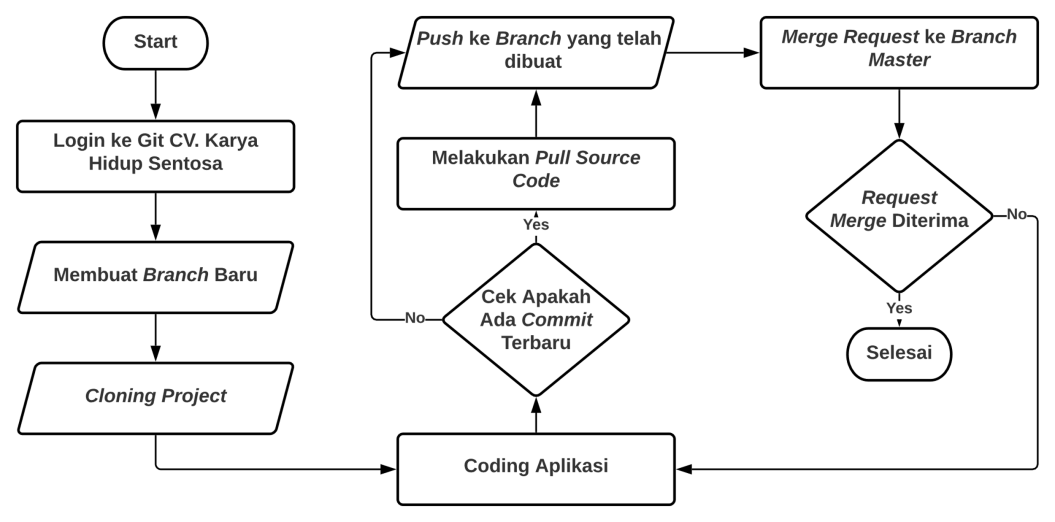
1. **Produk PKL 2 : Penggunaan Git di Lingkup Perusahaan**
2. Penjelasan Singkat Jenis Pekerjaan atau Produk

Dalam melakukan pembuatan aplikasi, kita dapat melakukan kolaborasi dengan orang lain maupun dalam kelompok. Dalam melakukan kegiatan tersebut terdapat *software* yang mendukung untuk terjalinnya kerja sama untuk melakukan kolaborasi bersama.

Salah satunya adalah Git, sebuah *version control system* yang digunakan para *programmer* untuk mengembangkan *software* secara bersama-bersama. Fungsi utama Git yaitu mengatur versi dari *sourcecode* program dengan memberikan tanda baris dan *code* mana yang ditambah atau diganti.

Git memudahkan *programmer* untuk mengetahui perubahan source *code*-nya daripada harus membuat *file* baru. Dengan Git kita tak perlu khawatir *code* yang kita kerjakan bentrok dengan *code* sesama tim, karena setiap *programmer*pastinya membuat *branch* sebagai workspace-nya.

1. Alat dan Bahan yang Digunakan
2. Komputer dan *peripheral*-nya
3. Sistem Operasi Windows 7
4. Visual Studio Code
5. *Browser* Chrome
6. Gitbash
7. Keselamatan Kerja
8. Niat dan berdoa sebelum melakukan kegiatan.
9. Memastikan semua alat dan bahan yang diperlukan sudah tersedia.
10. Menggunakan peralatan sesuai fungsinya.
11. Gunakan internet sebagaimana mestinya.
12. Posisi duduk tegap.
13. Memperhatikan jarak pandang mata dengan monitor minimal 30 cm.
14. Kurangi kecerahan monitor supaya nyaman untuk mata.
15. Hindari makan dan minum di dekat monitor.
16. Matikan perangkat bila sudah tidak digunakan.
17. Bersihkan perangkat yang digunakan bila kotor.
18. Gambar Kerja

****

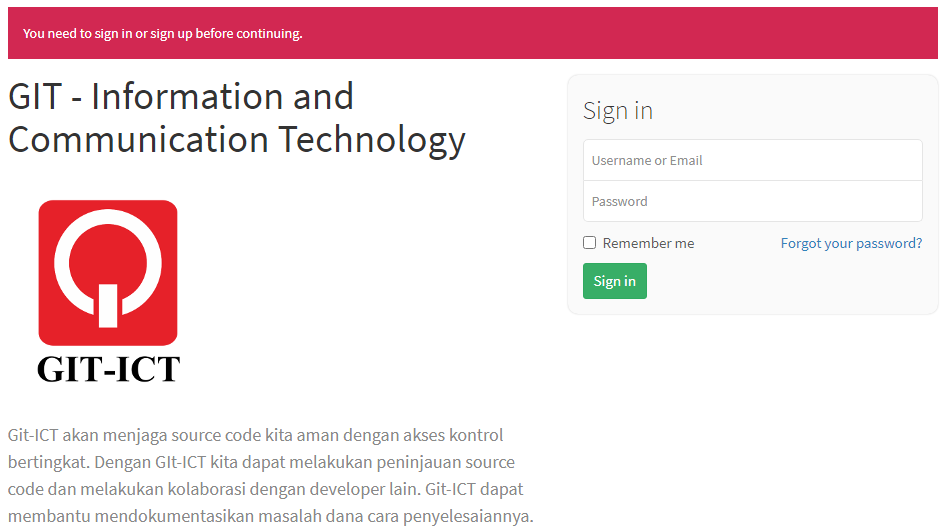
Gambar 3.83. Gambar Kerja Penggunaan Git

1. Analisis

Didalam pembuatan dan pengembangan aplikasi baik web maupun android pastinya dibuat tim supaya pembuatan dan pengembangan aplikasi lebih cepat dan tidak memakan banyak waktu. Seorang *programmer* pastinya ada beberapa bahasa pemrograman yang tidak mereka kuasai, beberapa ada yang tidak ahli dalam mendesain tampilan aplikasi, tidak ahli dalam mengelola *database*, dan lain-lain.Maka dari itu dibentuknya tim sangat penting dalam melengkapi kekurangan tersebut.

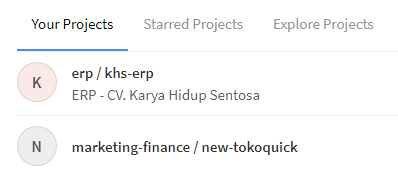
Saat para *programmer* dalam tim sudah mulai men-develop aplikasi, muncul masalah baru yaitu bagaimana cara menggabungkan *source code* dari beberapa *programmer* tersebut. Maka dari itu Git menjawab pertanyaan dan mengatasi masalah tersebut. Dengan Git, berapa banyakpun *programmer* yang ada mereka bisa saling berkolaborasi antar *source code* meraka, sehinggan program menjadilebih terstrukur, tertata, dan mudah dipahami sesama *programmer*.

1. Langkah Kerja
2. Login ke Git Quick
3. Buka halama Git Quick menggunakan *browser*. *Login* dengan menggunakan username dan password yang telah didaftarkan oleh atasan.



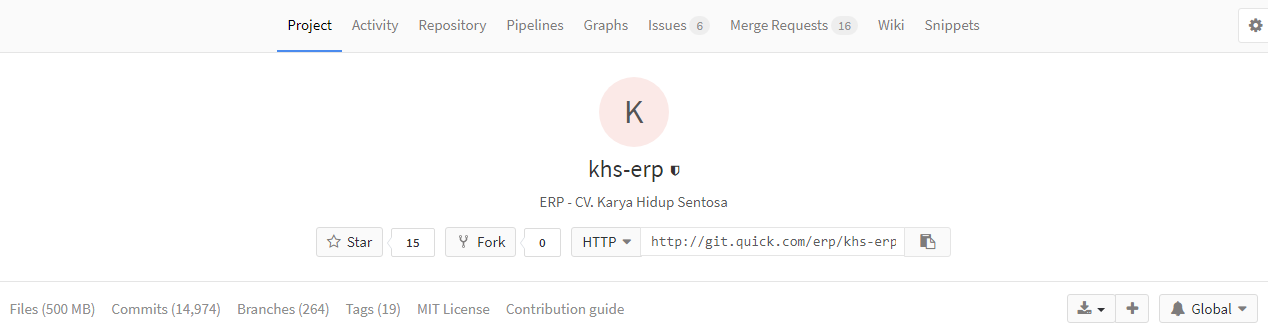
Gambar 3.84. Halaman Login Git Quick

1. Maka akan muncul daftar project, daftar ini berdasarkan project yang diikuti oleh user.



Gambar 3.85. Daftar Project Git

1. Membuat branch baru di Git Quick
2. Buka salah satu project. Disini penulis membuka *projecterp/khs-erp*.
3. Anda akan memasuki halaman awal *project erp/khs-erp*.



Gambar 3.86. Halaman Git Quick

1. Kemudian klik tab *Branches*.



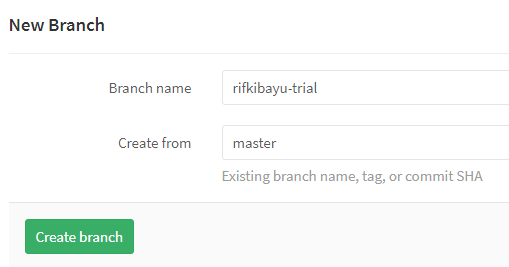
Gambar 3.87. Tab Pada Menu Project

1. Klik *New Branch* untuk beralih ke halaman yang berfungsi untuk membuat *­branch*baru.



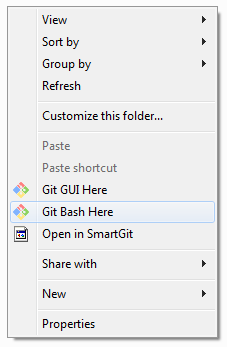
Gambar 3.88. Tombol New *branch*

1. Untuk membuat *branch* baru, ketikkan nama *­branch* di kolom *inputBranchname*. Lalu pilih *branch* lain sebagai duplikasi untuk dimasukkan ke *branch* yang akan dibuat. Biasanya akan otomatis memilih master.



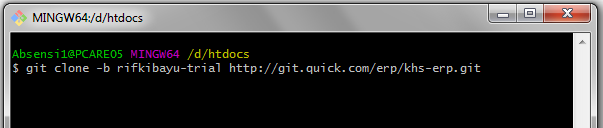
Gambar 3.89. *Form* Buat *Branch* Baru

1. Jika sudah klik button *Create Branch* untuk memproses pembuatan branch baru.
2. Clone branch ke directory local
3. Buka directory *web server* local menggunakan *file explorer*.
4. Buka gitbash yang mengarah ke *directory* tersebut, dengan cara klik kanan pada *directory* tersebut.



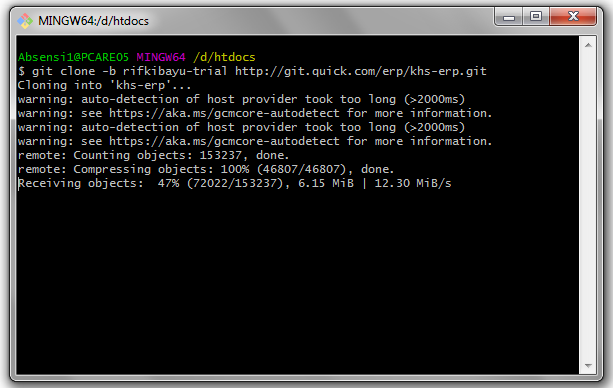
Gambar 3.90. Pilihan Membuka Gitbash Pada Direktori

1. Kemudian jalankan baris perintah seperti berikut.



Gambar 3.91. *Code* Untuk *CloneBranch*

1. Script yang dijalankan, *git clone –b nama\_branch link\_repository*, tekan enter untuk menjalankannya.
2. Kemudian tunggu proses *clone branch* hingga berhasil.



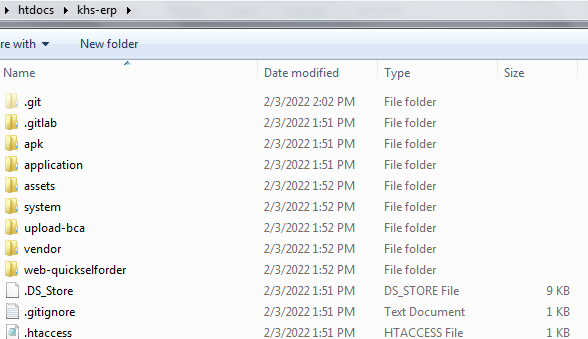
Gambar 3.92. Proses *CloneBranch*

1. Push *commit* ke branch
2. Buka folder *repository* yang berhasil diclone tadi melalui Visual Studio Code.



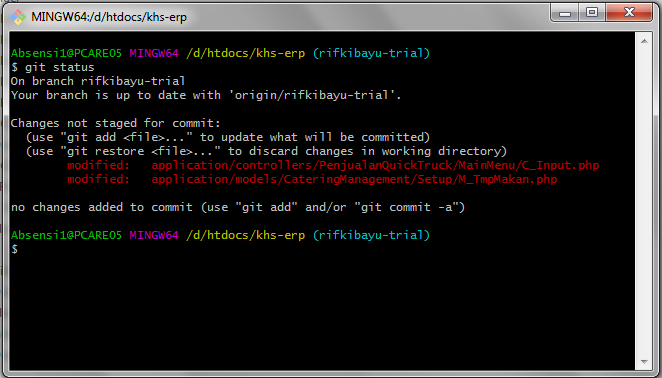
Gambar 3.93. Project Pada Visual Studio Code

1. Coba ubah salah satu file yang berada pada *directory* tersebut menggunakan *code editor.*
2. Jika sudah berhasil diubah, buka *directory* hasil clone tadi menggunakan *file explorer*.



Gambar 3.94. Project Hasil Clone

1. Lantas buka gitbash yang mengarah ke *directory* tersebut.
2. Jalankan perintah *git status*, untuk mengecek perubahan apa saja yang telah terjadi.



Gambar 3.95. Perubahan Pada *Branch*

1. Terdapat dua file yang termodifikasi.
2. Disini penulis hanya akan commit salah satu file saja. Menggunakan perintah *git add lokasi\_file\_modifikasi*.



Gambar 3.96. Menambahkan File *Commit* Git

1. Jalankan perintah *git status* kembali untuk mengecek apakah file berhasil dicommit atau belum. File yang tercommit akan berada di daftar *commit* dan berwarna hijau.



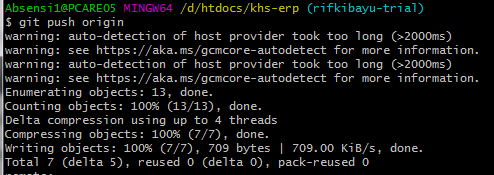
Gambar 3.97. Status *CommitBranch*

1. Kemudian tambahkan keterangan *commit* dengan perintah *git commit –m “keterangan”*.



Gambar 3.98. Keterangan *Commit*

1. Jika sudah, langkah terakhir jalankan perintah *git push origin* untuk push ke branch.



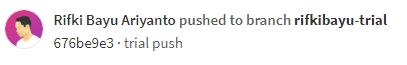
Gambar 3.99. Push *Commit* Ke *Branch*

1. Open merge request ke branch master Git Quick
2. Buka Git Quick kembali, buka project sesuai dengan *branch* yang telah dipush tadi.
3. Sebelum di merge, periksa terlebih dahulu dengan membuka *tab activity*.



Gambar 3.100. Menu Git Quick

1. Cari activity push tadi, jika ketemu klik pada nama *branch*.



Gambar 3.101. History Git Quick

1. Pilih *commit* yang sesuai dengan keterangan pada saat push *commit* tadi.



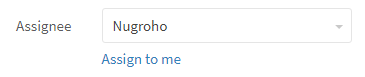
Gambar 3.102. *Commit* Pada *Branch*

1. Cek kembali file file yang telah dimodifikasi tadi.
2. Kembali ke *tab activity*, klik button *Create Merge Request* di pojok atas kanan.



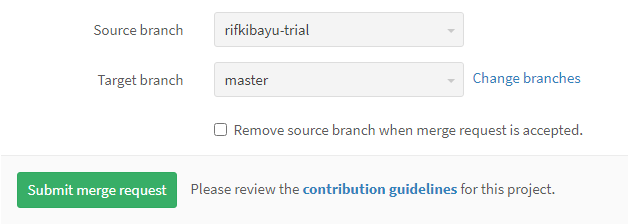
Gambar 3.103. Tombol *Merge* Request

1. Kemudian pilih nama atasan yang akan menyetujui request merge.



Gambar 3.104. Target Approve *Merge*

1. Jika sudah klik tombol *Submit merge request* untuk meminta pesetujuan merge ke atasan yang dipilih tadi.



Gambar 3.105. *FormMerge* Request

1. Tunggu diapprove oleh atasan.
2. Pelaksanaan Quality Control

Setelah melakukan develop suatu aplikasi baik membuat aplikasi baru atau penambahan fitur, seorang *programmer* CV. Karya Hidup Sentosa harus memberitahu atasan terlebih dahulu tentang kelanjutan dari aplikasi yang telah dikembangkan atau telah dibuat.

Biasanya atasan akan memberitahu kepada *programmer*, seperti melakukan *merge* aplikasi ke *branch masterserver* Git Quick, penambahan fitur baru, pengecekkan *source code*, dan developaplikasi tersebut ditunda terlebih dahulu karena suatu hal yang lebih urgent.

Yang harus *programmer* lakukan untuk kemungkinan yang akan terjadi dari perintah atasan tersebut:

1. Jika atasan memerintahkan melakukan *merge* aplikasi ke *branch master server* Git Quick, maka lakukan mergerequest ke *branch master server* Git Quick. Sesuaikan project yang dikerjakan, jika projecttersebut termasuk project *erp* maka merge ke projecterp.
2. Jika atasan memerintahkan melakukan penambahan fitur baru, maka bertanyalah terlebih dahulu kepada atasan kejelasan dari fitur baru yang akan dibuat supaya develop aplikasi menjadi lebih terarah dan jelas, baru kemudian melakukan coding.
3. Jika atasan memerintahkan kita untuk melakukan pengecekkan *source code*, bisanya atasan menyuruh *programmer* senior untuk melakukan pengecekan *source code*. Akan tetapi sebelum menyerahkan ke senior, kita coba cek sendiri terlebih dahulu.
4. Jika atasan memerintahkan kita untuk menunda develop aplikasi yang dikerjakan, maka kita berhenti untuk develop aplikasi tersebut. Lalu menunggu perintah atasan selanjutnya.
5. Kesimpulan

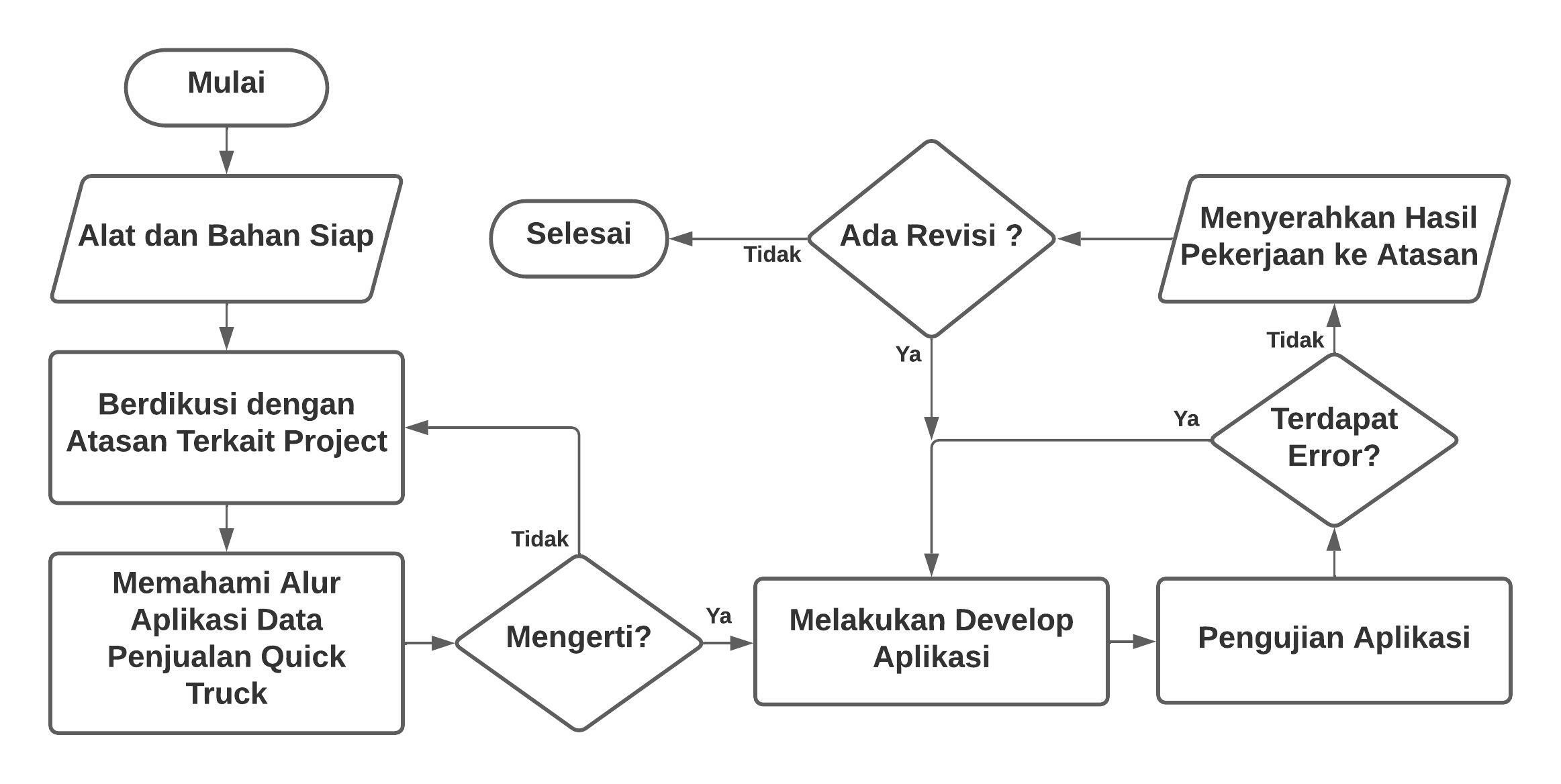
Git adalah *softwareversion controlling* yang mana setiap perubahan *source code* pada sebuah aplikasi dapat dilihat perubahan perubahan yang terjadi, selain itu Git juga memudahkan dalam berkolaborasi dengan orang lain karena memiliki sistem yang terpusat dan terkendali yang tentunya keamanan *source code* lebih aman.

1. **Produk PKL 3 : Pembuatan Aplikasi Data Penjualan Quick Truck**
2. Penjelasan Singkat Jenis Pekerjaan atau Produk

Aplikasi Data Penjualan Quick Truck merupakan aplikasi yang digunakan untuk mengurusi data-data penjualan produk quick truck. Aplikasi digunakan menginputkan data penjualan, input pre order, monitoring data-data yang telah berhasil diinputkan, hingga mengexport data ke dalam format file.

Aplikasi ini memiliki lima menu, yaitu master group, master customer, input & report data penjualan, input data inquiries, dan progress rencana penjualan. Menu master group dan customer digunakan untuk mengatur data pelanggan quick truck. Menu input & report data penjualan digunakan untuk menginputkan data penjualan quick truck sekaligus digunakan untuk monitoring data penjualan quick truck. Dan dua menu terakhir digunakan untuk input data pre order serta untuk monitoring data pre order.

1. Alat dan Bahan yang Digunakan
2. Komputer dan *peripheral*-nya
3. Sistem Operasi Windows 7
4. Visual Studio Code
5. *Browser* Chrome
6. Keselamatan Kerja
7. Niat dan berdoa sebelum melakukan kegiatan.
8. Memastikan alat dan bahan yang diperlukan sudah tersedia.
9. Menggunakan peralatan sesuai fungsinya.
10. Gunakan internet sebagaimana mestinya.
11. Posisi duduk tegap.
12. Memperhatikan jarak pandang mata ke monitor minimal 30 cm.
13. Kurangi kecerahan monitor supaya nyaman untuk mata.
14. Hindari makan dan minum di dekat monitor.
15. Matikan perangkat bila sudah tidak digunakan.
16. Bersihkan perangkat yang digunakan bila kotor.
17. Gambar Kerja

****

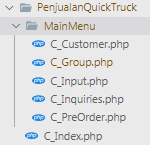
Gambar 3.106. Gambar Kerja Aplikasi Data Penjualan Quick Truck

1. Analisis

Aplikasi Data Penjualan Quick Truck akan mempermudah dalam menginputkan data-data penjualan maupun pre order, karena metode yang sebelumnya digunakan masih menggunakan metode menyimpannya di file excel. Dengan dibuatnya aplikasi khusus yang digunakan untuk mengurus data-data tersebut tentu akan meringkas waktu dan data akan tersimpan lebih aman dalam database. Tidak hanya input tetapi aplikasi ini memiliki fitur export data ke file excel maupun pdf. Sehingga data yang disimpan dalam *database* juga dapat diubah ke file excel maupun pdf.

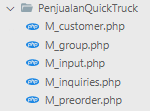
1. Langkah Kerja
2. Setelah sebelumnya telah melakukan *clone branch* dari remote *repository* Quick ERP pada Git CV. Karya Hidup Sentosa, langkah pertama yang dilakukan adalah mengganti koneksi ke *Database* Quick Dev. *Database* Quick Dev ini dikhususkan untuk *programmer* dalam melakukan develop aplikasi, sebelum nantinya dipublikasikan dan digunakan oleh user.
3. Kemudian buat file *controller*, *model*, dan *view*. *View* merupakan bagian yang berisikan html untuk tampilan, dan *Model* merupakan bagian yang berisikan query query untuk mengeksekusi database. Sedangakan *Controller* merupakan penghubung antara *View* dengan *Model*. Lokasi untuk menyimpan Controller ada di “.../application/controller/”, lokasi View berada di “.../application/view/”, dan lokasi model berada di “.../application/model/”.

Struktur folder dan file *Controller*:



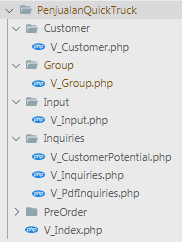
Gambar 3.107. Struktur *Controller* Aplikasi Quick Truck

Struktur folder dan file *Model*:



Gambar 3.108. Struktur *Model* Aplikasi Quick Truck

Struktur folder dan file *View*:



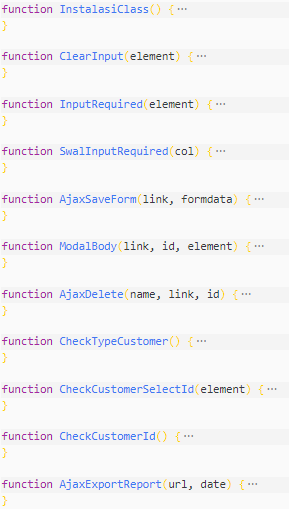
Gambar 3.109. Struktur *View* Aplikasi Quick Truck

1. Langkah selanjutnya adalah menambahkan konfigurasi *routes* pada *directory* “.../application/config/routes”. Fungsi dari *routes* sendiri adalah sebagai penghubung antara url dengan *controller*. *Routes* yang dipakai:



Gambar 3.110. *Routes* Aplikasi Quick Truck

1. Sebelum memulai membuat menu, buat sebuah file *javascript* yang nantinya akan bakal sering dipakai kedepannya. Buat file customDPQT.js di *directory* “.../assets/js/”.
2. Buat *function*-*function* pada *javascript* yang nantinya akan dipakai untuk menyimpan data *form*, untuk menampilkan popup untuk delete data, dan untuk melakukan setup select.



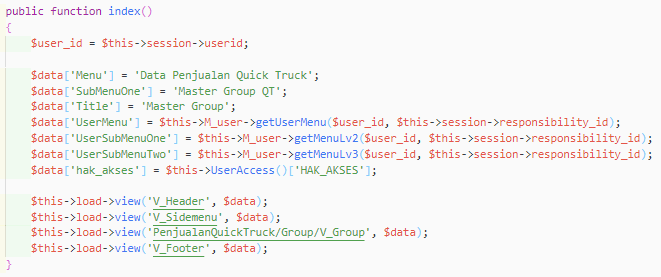
Gambar 3.111. *Function JavaScript* Aplikasi Quick Truck

1. Membuat menu Master Group, *routes* yang dipakai:



Gambar 3.112. *Routes* Menu Group

1. Buka file C\_Group yang letaknya sesuai dengan struktur folder dan file diatas tadi.
2. Buat *function* index yang berisikan baris kode untuk memanggil *view* V\_Group.



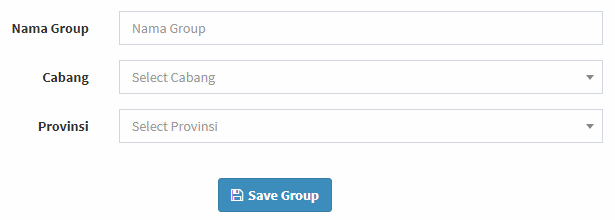
Gambar 3.113. Kode Memanggil *View* Menu Group

1. Langkah selanjutnya buka file V\_Group untuk membuat tampilan yang berisikan kolom input group.



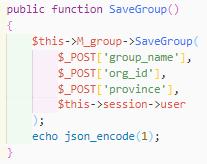
Gambar 3.14. Kode *Html View* Menu Group

1. Yang nantinya jika url “.../PenjualanQuickTruck/Group” diakses maka akan menghasilkan tampilan seperti ini.



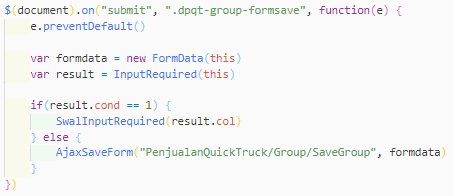
Gambar 3.115. Tampilan *View* Menu Group

1. Kembali ke file *controller* C\_Group, buat function SaveGroup yang nantinya akan digunakan untuk menangkap data dari *View* dan menyimpan ke *database* melalui *Model*.



Gambar 3.116. *Function* Menyimpan Group

1. Buat *event* pada file *javascipt* customDPQT, event jika *form* disubmit. Data akan dilanjutkan ke *controllerfunction* SaveGroup tadi.



Gambar 3.117. *Event* Submit *Form* Menu Group

1. Untuk fitur menyimpan berhasil dibuat, langkah selanjutnya adalah membuat table untuk menampilkan group yang telah berhasil disimpan tadi. Metode yang digunakan untuk menampilkan data pada tabel adalah *serverside*. Dengan menggunakan metode *serverside*, halaman akan menjadi lebih ringan karena data yang diambil dari database adalah per page/halaman. *Plugins* yang dipakai adalah *pluginsjavascript* datatable.
2. Buka file V\_Group tambahkan baris *html* untuk membuat sebuah tabel.



Gambar 3.118. Kode *Html* Tabel Menu Group

1. Masih di file V\_Group tambahkan baris script untuk meload data secara *serverside* menggunakan datatable.



Gambar 3.119. *AjaxServerSide* Tabel

1. Buka file C\_Group buat *function* ServerSideGroup, *function* ini digunakan untuk mingirimkan data ke *View*.



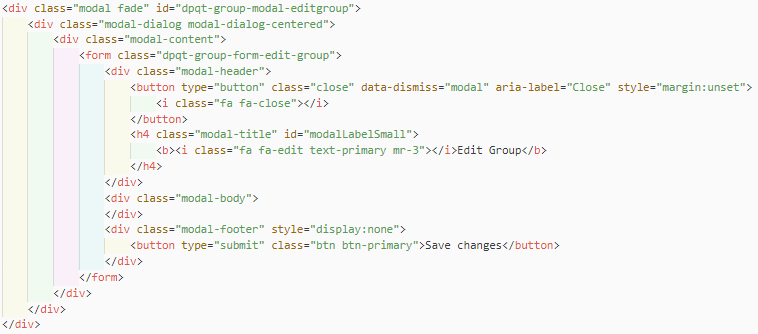
Gambar 3.120. *FunctionServerSide* MenuGroup

1. Coba refresh / akses url “.../PenjualanQuickTruck/Group” maka pada tabel akan terdapat data yang pernah disimpan.



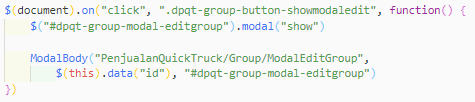
Gambar 3.121. Tampilan Tabel Menu Group

1. Pada tabel diatas terdapat dua button pada kolom action, yaitu button edit/hijau dan button delete/merah.
2. Buka file V\_Group, tambahkan baris*html* untuk membuat tampilan pop up.



Gambar 3.122. *Html* Modal Edit Group

1. Buat *event* pada file *javascipt* customDPQT.js, *event* jika button edit terclick maka akan memunculkan pop up yang berisikan *form* edit.



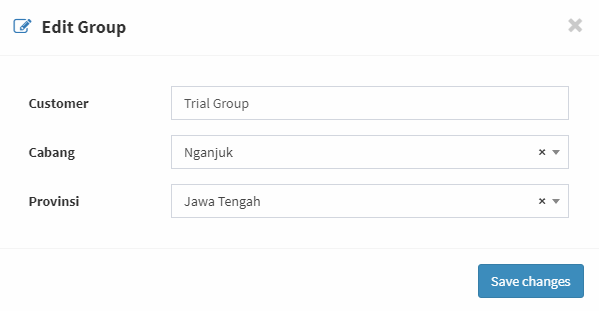
Gambar 3.123. *Event* Klik Tombol Edit Group

1. Buka file C\_Group buat *function* ModalEditGroup, *function* ini digunakan untuk mengisi data data pop up.



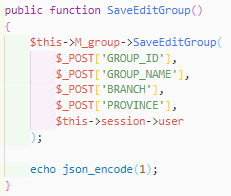
Gambar 3.124. *FunctionForm* Edit Group Modal

1. Coba klik salah satu button edit ditable, maka akan muncul pop up edit group.



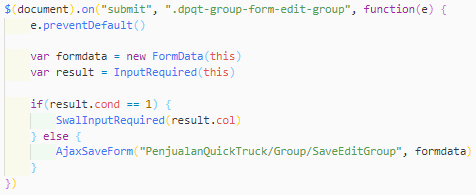
Gambar 3.125. Tampilan Modal Edit Group

1. Buat *function* lagi, *function* SaveEditGroup digunakan untuk menyimpan update data melalui pop up.



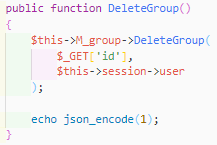
Gambar 3.126. *Function* Menyimpan Edit Group

1. Buat *event* pada file javasciprt customDPQT.js, *event* jika *form* popup tersubmit.



Gambar 3.127. *Event* Submit *Form* Edit Group Modal

1. Fitur edit berhasil dibuat, selanjutnya buat fitur delete group. Buka kembali C\_Group, buat *function* DeleteGroup.



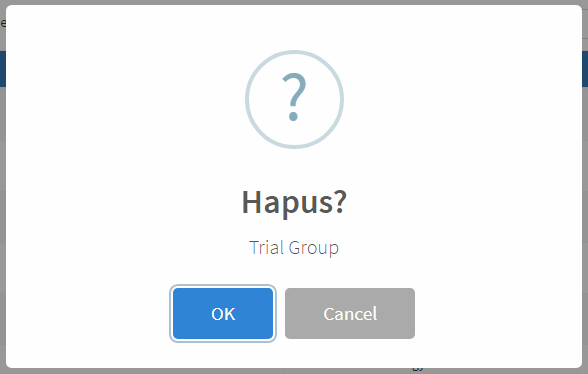
Gambar 3.128. *Function* Hapus Group

1. Buat *event* pada file *javascript* customDPQT.js, *event* jika button delete diclick maka akan muncul Verifikasi kembali untuk menghapus.



Gambar 3.129. *Event* Klik Tombol Hapus Group

1. Coba click salah satu button delete pada table, maka akan muncul pop up peringatan.



Gambar 3.130. Tampilan Pop-up Hapus Group

1. Menu Master Group telah selesai dibuat.
2. Membuat menu Master Customer, *routes* yang dipakai:



Gambar 3.131. *Routes* Menu Customer

1. Buka file C\_Customer, buat *function* index yang nanti akan digunakan untuk memanggil *view* V\_Customer.



Gambar 3.132. *Function* Memanggil *View* Menu Customer

1. Kemudian buka file *view* V\_Customer, buat code*html* untuk membuat *form* input customer.

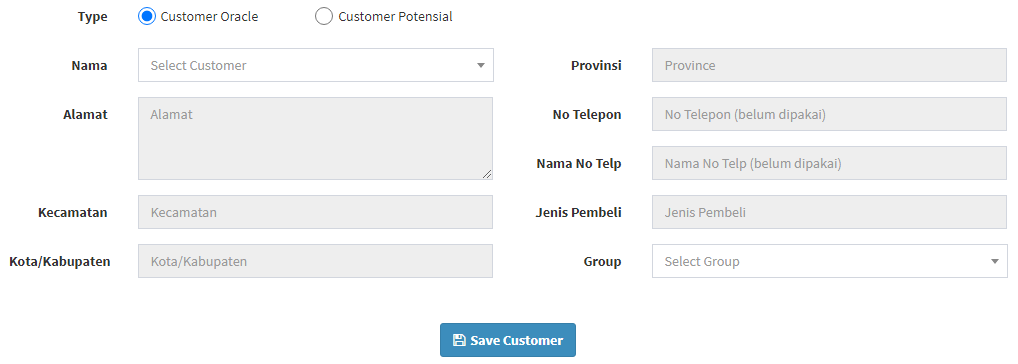


Gambar 3.133. *HtmlView* Menu Customer



Gambar 3.134. *HtmlView* Menu Customer

1. Nantinya jika url “.../PenjualanQuickTruck/Customer” diakses maka akan menghasilkan tampilan seperti ini.



Gambar 3.135. Tampilan Menu Customer

1. Buat *event* pada *javascript* customDPQT.js, *event* jika kolom nama terisi, maka otomatis kolom kolom dibawahnya akan terisi juga.



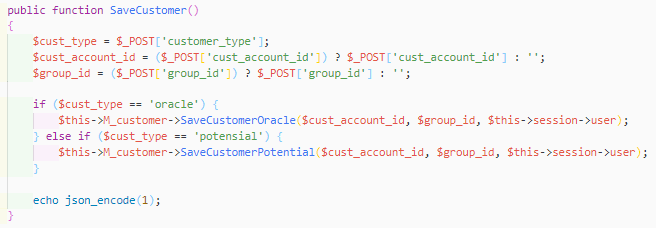
Gambar 3.136. *Event* Kolom Customer Berubah

1. Kemudian buat *event*lagi, *event* jika *form* tersubmit.



Gambar 3.137. *Event* Submit *Form* Menu Customer

1. Lantas buka file C\_Customer, buat *function* SaveCustomer yang akan digunakan menyimpan data ke *database*.



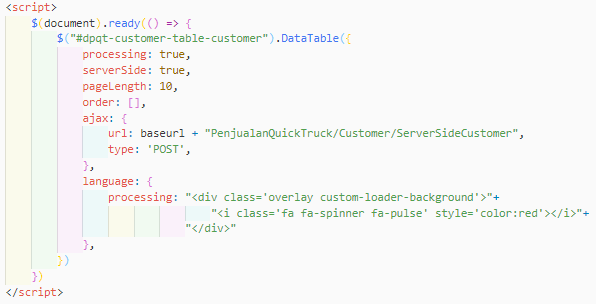
Gambar 3.138. *Function* Menyimpan Customer

1. Fitur input customer sudah berhasil dibuat, sekarang buka file V\_Customer dan tambahkan *html* untuk membuat tabel yang menampilkan data customer yang berhasil disimpan.



Gambar 3.139. *Html* Tabel Menu Customer

1. Masih di V\_Customer, tambahkan script *serverside* untuk menampilkan data ke tabel.



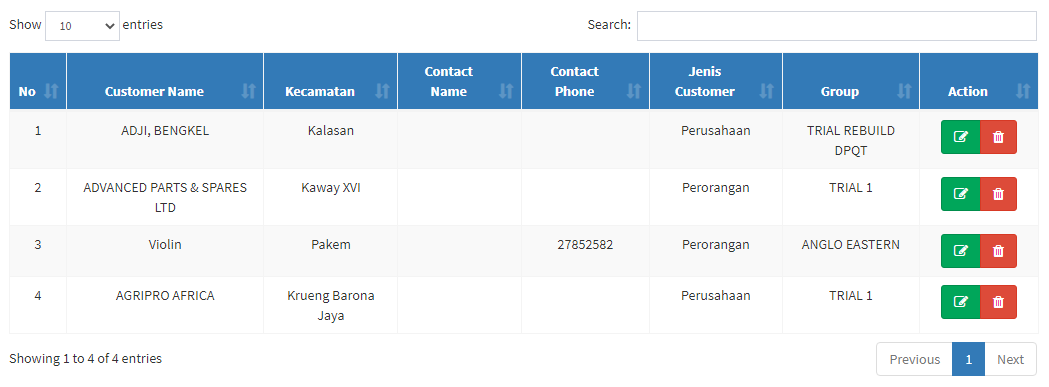
Gambar 3.140. *AjaxServerSide* Tabel Customer

1. Kembali lagi ke file C\_Customer, buat *function* ServerSideCustomer yang digunakan untuk mengirim data dari *database* ke table secara *serverside* melalu *ajax* datatable.



Gambar 3.141. *FunctionServerSide* Tabel Customer

1. Sekarang coba akses url menu Master Customer “.../PenjualanQuickTruck/Customer”, jika tidak terdapat error maka *serverside*tabel berhasil dibuat. Tampilan tabel seperti berikut.



Gambar 3.142. Tampilan Tabel Customer

1. Di tabel terdapat dua tombol, button edit dan button delete. Selanjutnya buat kedua fitur tersebut.
2. Buat fitur update terlebih dahulu, buat *event* pada *javascript* customDPQT.js. Event jika button edit yang berwarna hijau diclick maka akan memunculkan pop up.



Gambar 3.143. *Event* Klik Tombol Edit Customer

1. Kemudian buka file V\_Customer, tambahkan baris *html* untuk membuat tampilan pop up yang nantinya akan digunakan untuk *form* edit customer.



Gambar 3.144. *HtmlView* Modal Edit Customer

1. Beralih ke file C\_Customer, buat *function* ModalEditCustomer untuk menampilkan *form* beserta data customer yang akan dimunculkan ke pop up nantinya.



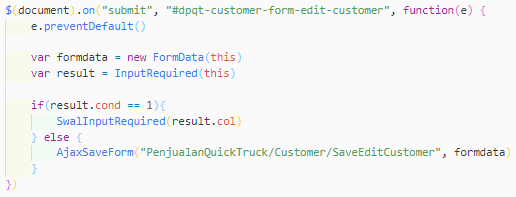
Gambar 3.145. *FunctionForm* Data Modal Edit Customer

1. Coba klik button edit, jika pop up sudah muncul beserta *form*data berarti berhasil untuk memunculkan pop up.



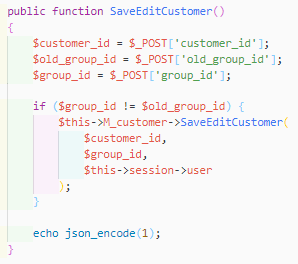
Gambar 3.146. Tampilan Pop-up Edit Customer

1. Buat *event* pada *javascript* customDPQT.js jika *form* edit tersubmit.



Gambar 3.147. *Event* Submit *Form* Edit Customer

1. Beralih ke C\_Customer, buat *function* SaveEditCustomer untuk menyimpan data edit dari *form* pop up.



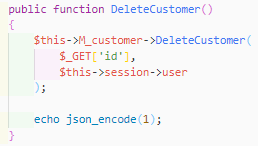
Gambar 3.148. *Function* Menyimpan Edit Customer

1. Fitur edit customer berhasil dibuat, sekarang buat fitur delete customer.
2. Buat *eventjavascript* customDPQT.js, *event* click button delete akan memunculkan pop up persetujuan delete.



Gambar 3.149. *Event* Klik Tombol Hapus Customer

1. Buka kembali file C\_Customer, buat *function* DeleteCustomer. Nantinya akan diteruskan ke *model* untuk diproses pada *database*.



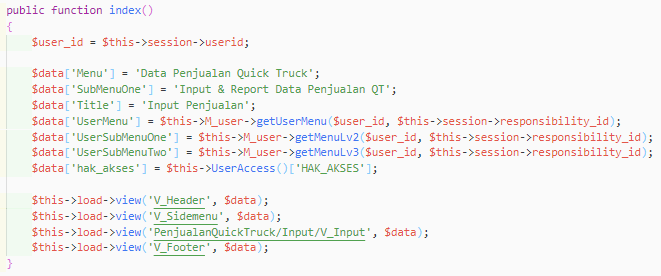
Gambar 3.150. *Function* Hapus Customer

1. Coba click button delete, jika pop up yang berisikan peringatan sudah muncul dan berjalan dengan lancar berarti fitur delete berhasil dibuat.
2. Membuat menu Input & Report Data Penjualan, *routes* yang dipakai:



Gambar 3.151. *Routes* Menu Input Data Penjualan

1. Buka file C\_Input, buat *function* index untuk memanggil *view* V\_Input.



Gambar 3.152. *Function* Memanggil *View* Menu Input

1. Buka V\_Input, tambahkan baris *html* untuk menampilkan *form* input data penjualan.

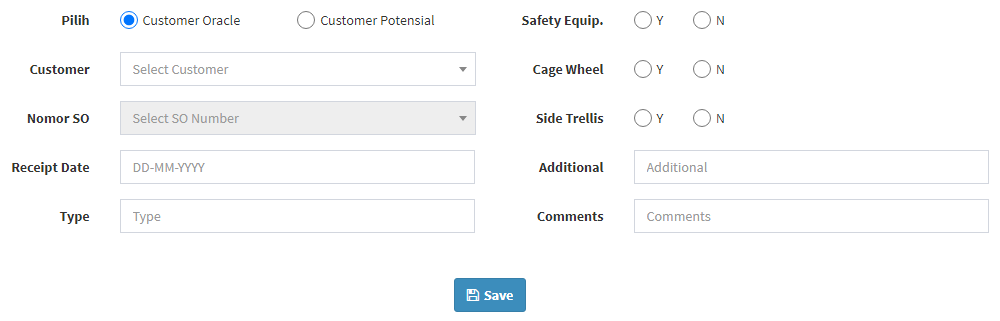


Gambar 3.152. *Html View* Menu Input



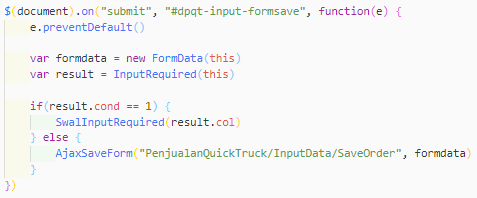
Gambar 3.153. *Html View* Menu Input

1. Akses url “.../PenjualanQuickTruck/InputData”, maka akan tampil seperti ini.



Gambar 3.154. Tampilan *Form* Input

1. Buat *event* form submit pada *javascript* customDPQT.js.



Gambar 3.155. *Event* Submit *Form* Input

1. Lantas buka file C\_Input, buat *function* SaveOrder untuk menangkap data dari *ajax* kemudian meneruskannya ke *model* untuk disimpan.



Gambar 3.156. *Function* Menyimpan Form Input

1. Fitur input order berhasil dibuat, sekarang buka file V\_Input dan tambahkan baris *html* untuk membuat tabel yang akan menampilkan data order.



Gambar 3.157. *Html* Tabel Input

1. Dan tambahkan script untuk menampilkan data tabel secara *serverside* menggunakan *ajax* datatable.



Gambar 3.158. *AjaxServerSide* Tabel Input

1. Pada script diatas juga membuat dom datatable custom yang nantinya akan digunakan untuk menempatkan tombol export report data penjualan.



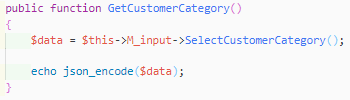
Gambar 3.159. DOM Datatable Tabel Input

1. buka file customDPQT.js tambahkan baris script untuk membuat tombol export report data penjualan.



Gambar 3.160. *Javascript* Buat Tombol Export

1. Buat*function* GetCustomerCategory untuk option diatas.



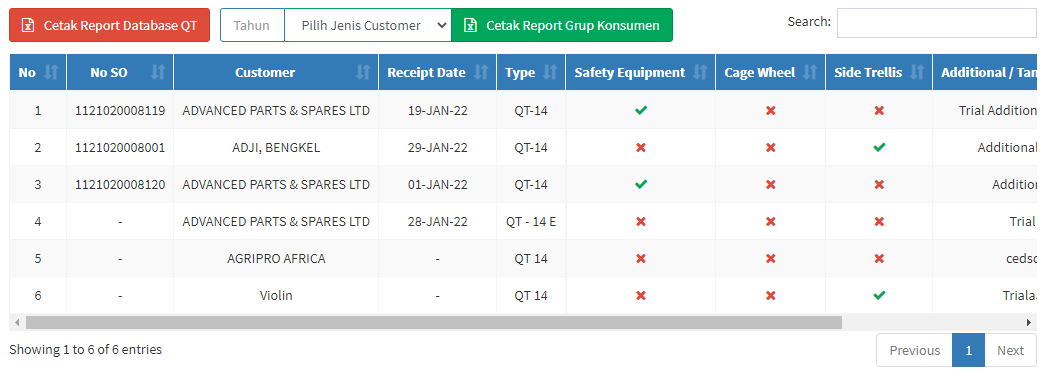
Gambar 3.161. *Function* List Kategori Customer

1. Buat juga *function* ServerSideOrder untuk menampilkan data pada table.



Gambar 3.162. *FunctionServerSide* Tabel Input

1. Akses url “.../PenjualanQuickTruck/InputData”, maka akan tampil tabel beserta datanya jika ada data pada database. Dan juga akan terdapat tombol report dikiri atas tabel.



Gambar 3.163. Tampilan Tabel Input

1. Buat *event* click button report database qt pada *javascript* customDPQT.js.



Gambar 3.164. *Event* Klik Tombol Export

1. Buka file C\_Input, buat *function* ExportExcelDatabaseQT. *Function* ini berisi baris *code* untuk membuat file excel menggunakan phpexcel.



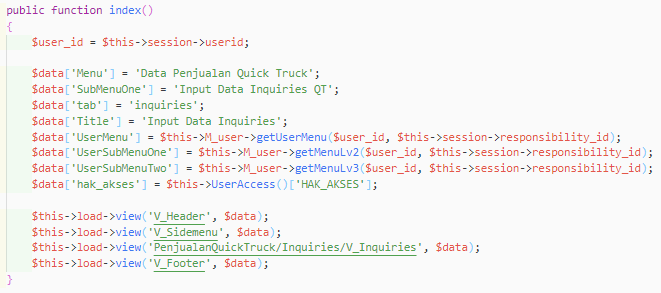
Gambar 3.165. *Function* Export Excel Data Input

1. Untuk tombol satunya juga hampir sama, bedanya menggunakan *event* submit bukan click karena menggunakan *form*.
2. Menu Input & Report Data Penjualan berhasil dibuat.
3. Membuat menu Input Data Inquiries, *routes* yang dipakai:



Gambar 3.166. *Routes* Menu Inquiries

1. Pada menu Input Data Inquiries terdapat 2 tab, yaitu Tab Input Inquiries dan tab Customer Potensial.
2. Tab Inquiries digunakan unuk menginputkan dan monitoring data inquiries dengan status inquiries.
3. Sedangkan tab customer potensial digunakan untuk memonitoring data dengan status no inquiries. Tab customer potensial tidak memiliki *form* input karena*form* input pada tab inquiries menginputkan data yang sama bedanya cuma pada statusnya.
4. Langkah pertama buat untuk tab Input Inquiries terlebih dahaulu. Buka file C\_Inquiries buat *function* index untuk memanggil *view* V\_Inquiries.



Gambar 3.167. *Function* Memanggil *View* Menu Inquiries

1. Buka file view V\_Inquiries, tambahkan baris *html* untuk membuat *form* input inquiries.*Form* ini digunakan untuk menginputkan data inquiries.



Gambar 3.168. *Html View* Menu Inquiries



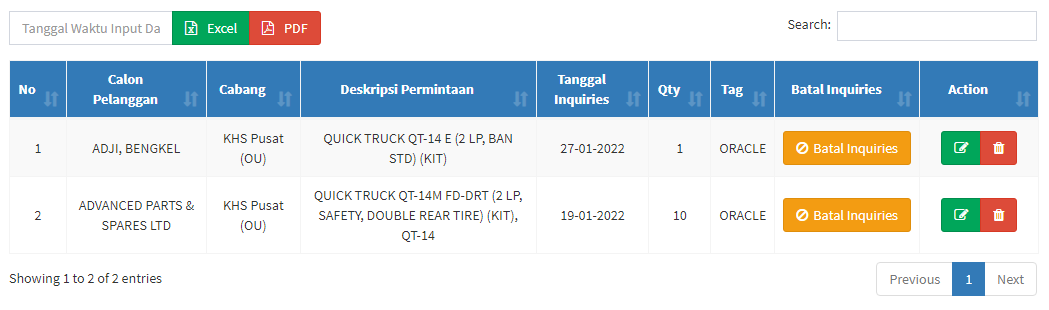
Gambar 3.169. *Html View* Menu Inquiries

1. Akses url “.../PenjualanQuickTruck/DataInquiries”, maka akan tampil seperti ini.



Gambar 3.170. Tampilan Menu *Form* Inquiries

1. Untuk proses save/menyimpan data sama dengan proses sebelum sebelumnya membuat *event* pada *javascript* dan membuat *function* pada controller, sehingga pada kali ini penulis tidak jelaskan.
2. Pada pembuatan tabel dan menampilkan data tabel juga sama dengan cara sebelumnya, bedanya kali ini pada tabel terdapat fitur edit, delete, button batal inquiries dan terdapat button export report yang menggunakan *form*.



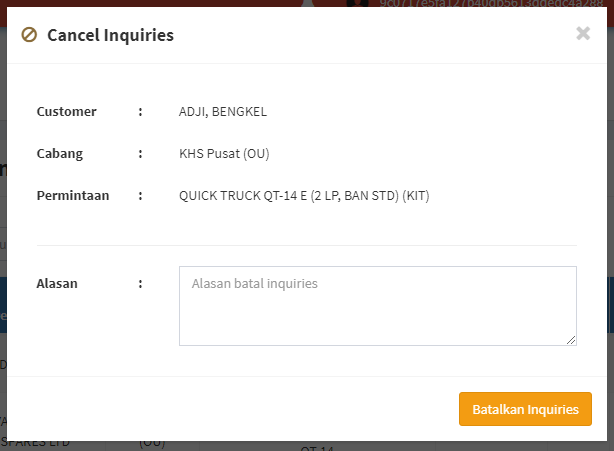
Gambar 3.171. Tampilan Tabel Inquiries

1. Untuk button Batal Inquiries yang berwarna kuning, merupakan button yang digunakan untuk merubah status INQUIRIES menjadi NO INQUIRIES. Maka seperti yang sudah penulis jelaskan tadi, tab customer potensial tidak perlu ada *form* input.
2. Buat *event* click button Batal Inquiries pada *javascript* customDPQT.js.



Gambar 3.172. *Event* Klik Tombol Cancel Inquiries

1. Jika button Batal Inquiries diclick maka akan muncul pop up seperti ini.



Gambar 3.173. Tampilan Pop-up Batal Inquiries

1. Buat *event* submit form cancel inquiries, caranya sama dengan cara edit sebelumnya jadi disini penulis tidak jelaskan. Dan bedanya pada *controller* dan *model* untuk menginputkannya saja.
2. Untuk button report yang berada diatas tabel caranya sama dengan cara sebelumnya, tetapi disini menggunakan *form* submit. Jadi buat *eventform* submit untuk report.



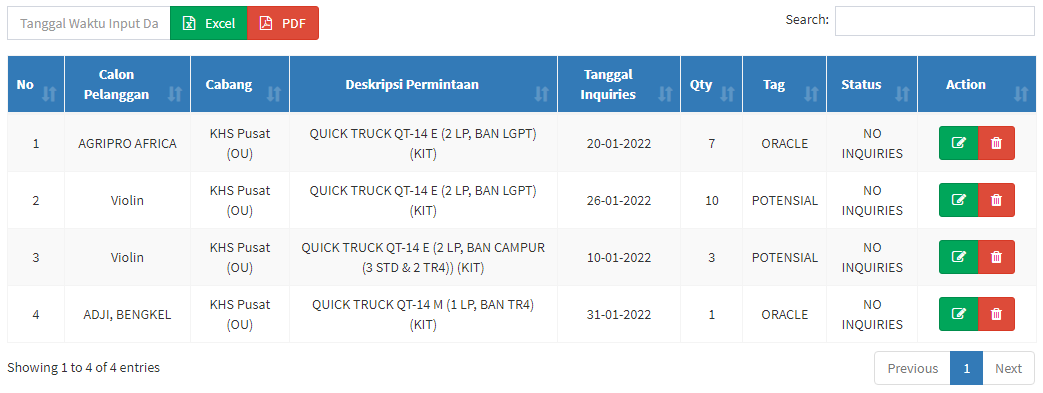
Gambar 3.174. *Event* Submit Export Data Inquiries

1. Dan buat *function* yang sama dengan cara sebelumnya bedanya kalau sebelumnya hanya untuk excel sekarang juga bisa untuk pdf. Untuk *function* export pdfnya seprti ini:



Gambar 3.175. *Function* Export PDF Inquiries

1. Untuk *view* pdfnya, buka file V\_PdfInquiries, dan tambahkan baris *html* untuk membuat tampilan sesuai format. Kalau excel tampilannya digenerate dari code phpexcel, sedangkan untuk pdf menggunakan mpdf yang memanggil *view*.
2. Untuk tab Input Inquiries sudah dibuat, untuk tab Customer isinya sebenarnya Cuma table yang sama persis dengan table di tab Input Inquiries. Perbedaannya kalau tabel di Input Inquiries menampilkan data dengan status INQUIRIES sedangkan tabel yang berada di tab Customer Potensial menampilkan data dengan status NO INQUIRIES. Dimana data tersebut diambil dari tabel database yang sama.



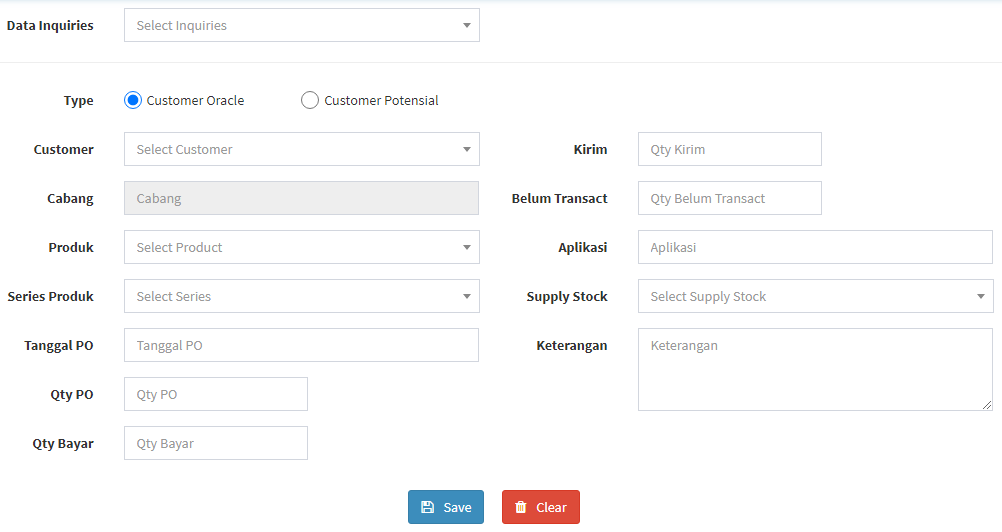
Gambar 3.176. Tampilan Tabel Customer Potensial

1. Untuk fitur export, edit, deletenya juga sama persis dengan tab Input Inquiries.
2. Menu Input Data Inquiries berhasil dibuat.
3. Menu Progress Rencana Penjualan juga mempunyai dua tab, yaitu tab Input Pre Order dan tab Data Series.
4. Membuat tab Input Pre Order menu Progress Rencana Penjualan, *routes* yang dipakai:



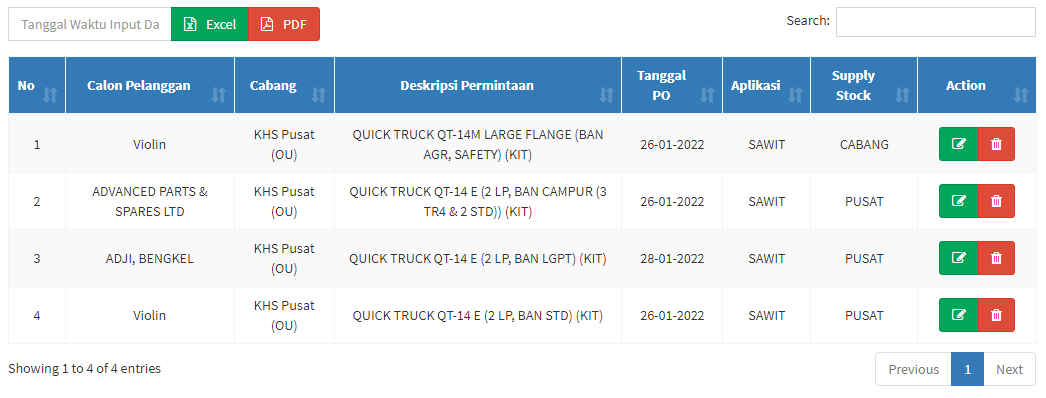
Gambar 3.177. *Routes* Menu Pre Order

1. Untuk tab Pre Order fiturnya sama dengan tab Input Inquires pada Menu Input Data Inquiries. Perbedaannya pada data apa saja yang diinputkannya. Selebihnya sama karena data yang diambil dan diinputkan juga dari tabel yang sama.



Gambar 3.178. Tampilan Menu *Form* Pre Order

1. Table pada tab Pre Order juga sama dengan tabel-tabel sebelumnya, cara mendapatkan data hingga fitur delete, update, dan export pun juga sama.



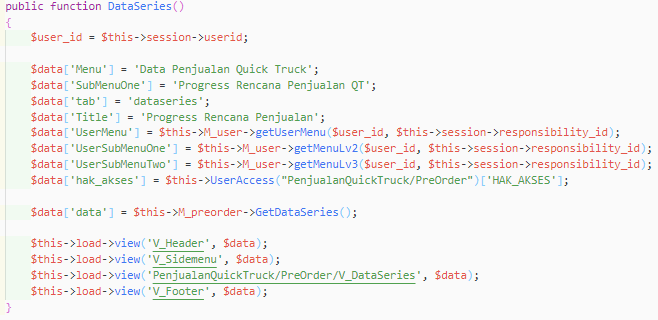
Gambar 3.179. Tampilan Tabel Menu Pre Order

1. Lantas untuk pembuatan tab Data Series, *routes* yang digunakan:



Gambar 3.180. *Routes* Tab Data Series

1. Buka file C\_PreOrder, buat *function* DataSeries yang akan digunakan untuk memanggil *view* V\_DataSeries.



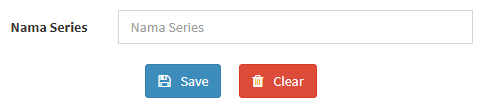
Gambar 3.180. *Function* Memanggil *View* Data Series

1. Kemudian buka V\_DataSeries, buat tampilan *form* input data series.



Gambar 3.181. *Html Form* Data Series

1. Akses url “.../PenjualanQuickTruck/DataSeries”, maka akan seperti ini.



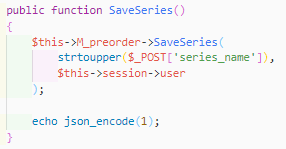
Gambar 3.182. Tampilan *Form* Data Series

1. Buat *even* submit *form* input data series pada *javascript* customDPQT.js.



Gambar 3.183. *Event* Submit *Form* Data Series

1. Buka C\_PreOrder kembali, buat *function* SaveSeries yang digunakan untuk menyimpan data series ke *database* melalui *model*.



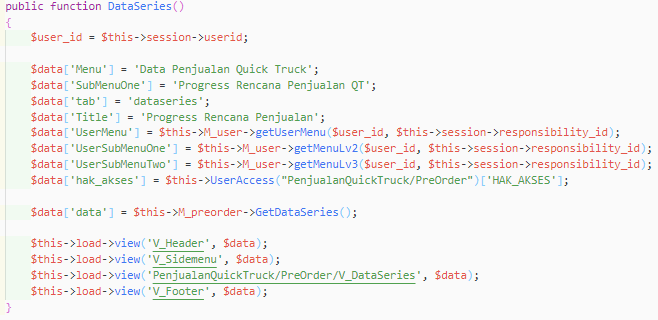
Gambar 3.184. *Function* Menyimpan Data Series

1. Kembali ke *view* V\_DataSeries, tambahkan *html* untuk membuat tabel yang akan menampilkan data series.



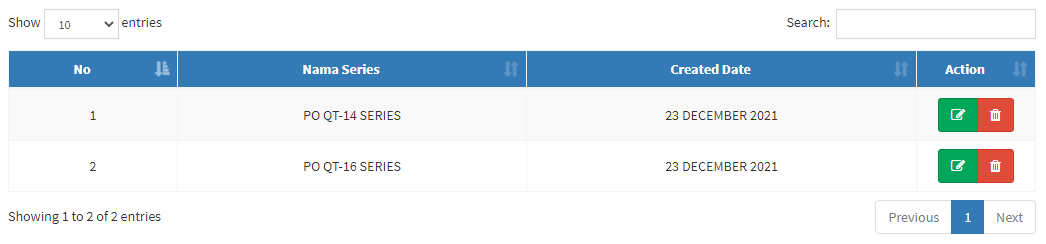
Gambar 3.185. *Html* Tabel Data Series

1. Cara diatas berbeda dengan cara-cara sebelumnya dalam menampilkan data tabel. Karen data series dikirim langsung melalui *function* DataSeries pada C\_PreOrder.



Gambar 3.186. Data *Array* Data Series

1. Tabel yang berhasil dibuat.



Gambar 3.187. Tampilan Tabel Data Series

1. Untuk fitur update dan delete sama dengan cara sebelum-sebelumnya.
2. Aplikasi Data Penjualan Quick Truck berhasil dibuat.
3. Pelaksanaan Quality Control

Setelah selesai membuat aplikasi Data Penjualan Quick Truck, langkah selanjutnya adalah untuk menguji coba secara mandiri terlebih dahulu. Pastikan program berjalan dengan lancar tanpa adanya *bug*maupun*error*.

Jika program dirasa telah sesuai dan tidak *bug* atau *error*, maka segera serahkan aplikasi ke atasan untuk dilakukan uji coba. Atasan biasanya akan berdiskusi dengan user yang akan menggunakan aplikasi tersebut. Tunggu keputusan atasan apakah adanya revisi atau perubahan alur. Jika atasan sudah merasa oke, tunggu perintah selanjutnya dari atasan.Biasanya atasan akan meberikan perintah untuk mengupload aplikasi ke server.

1. Kesimpulan

Aplikasi Data Penjualan Quick Truck merupakan aplikasi yang digunakan untuk memudahkan user dalam mengurusi data-data penjualan quick truck yang sebelumnya masih input melalui file excel. Selain input data-data penjualan aplikasi ini juga bisa digunakan untuk memonitoring data-data penjualan sekaligus membuat report file yang berupa excel maupun pdf.