

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.

A. Produk PKL 1 : Instalasi dan Konfigurasi Software yang Diperlukan

1. Penjelasan Singkat Jenis Pekerjaan atau Produk
 - a. 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*.

b. Visual Studio Code



Gambar 3.2. Logo Visual Studio Code

Visual Studio Code merupakan aplikasi *code editor* *cross-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.

c. 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 *Apache*, *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*.

d. 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*.

e. 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*.

f. 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).

g. Codeigniter



Gambar 3.7. Logo Codeigniter

Codeigniter merupakan salah satu *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 Codeigniter memiliki dokumentasi yang lengkap membuat codeigniter menjadi salah satu *framework* yang banyak digunakan.

2. Alat dan Bahan yang Digunakan

- a. Komputer dan *peripheral*-nya
- b. Sistem operasi Windows 7
- c. Browser
- d. Jaringan yang terkoneksi internet
- e. Sumber listrik

3. Keselamatan Kerja

- a. Niat dan berdoa sebelum melakukan kegiatan.
- b. Memastikan semua alat dan bahan yang diperlukan sudah tersedia.
- c. Menggunakan peralatan sesuai fungsinya.
- d. Gunakan internet sebagaimana mestinya.
- e. Posisi duduk tegap.
- f. Memperhatikan jarak pandang mata dengan monitor minimal 30 cm.
- g. Kurangi kecerahan monitor supaya nyaman untuk mata.
- h. Hindari makan dan minum di dekat monitor.
- i. Matikan perangkat bila sudah tidak digunakan.
- j. Bersihkan perangkat yang digunakan bila kotor.

4. GambarKerja



Gambar 3.8. Gambar Kerja Instalasi Software

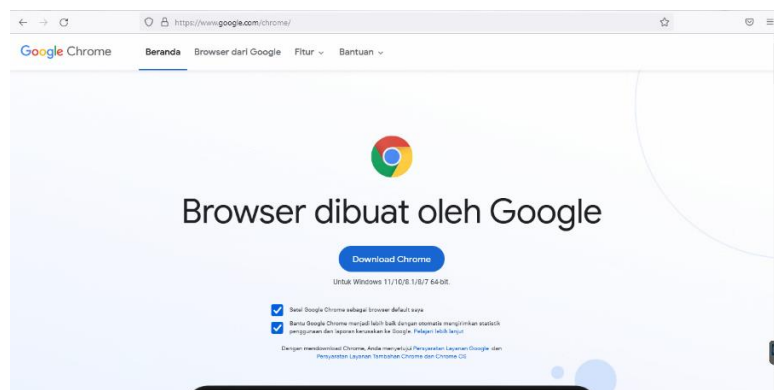
5. Analisis

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.

6. Langkah Kerja

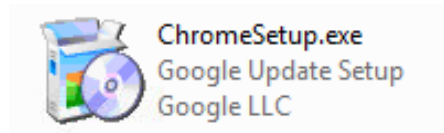
a. Instalasi Google Chrome

- 1) Buka halaman *website* resmi Google Chrome, <https://www.google.com/chrome/>.



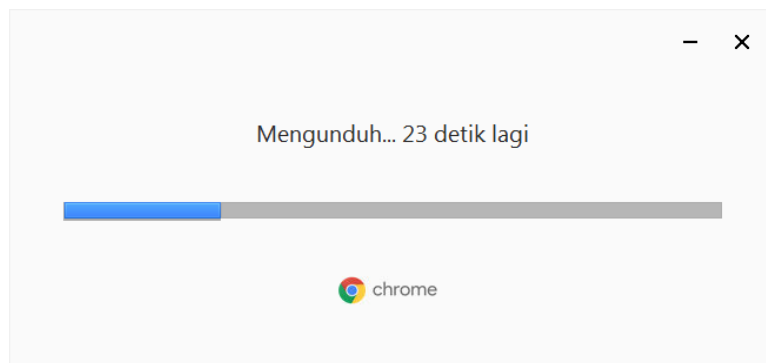
Gambar 3.9. Halaman *Website* Google Chrome

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



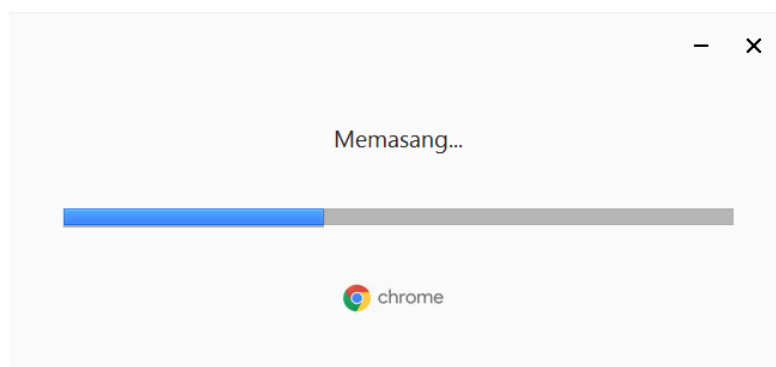
Gambar 3.10. *File Installer* Google Chrome

- 4) Tunggu program mengunduh aplikasi chrome.



Gambar 3.11. Setup Mengunduh Google Chrome

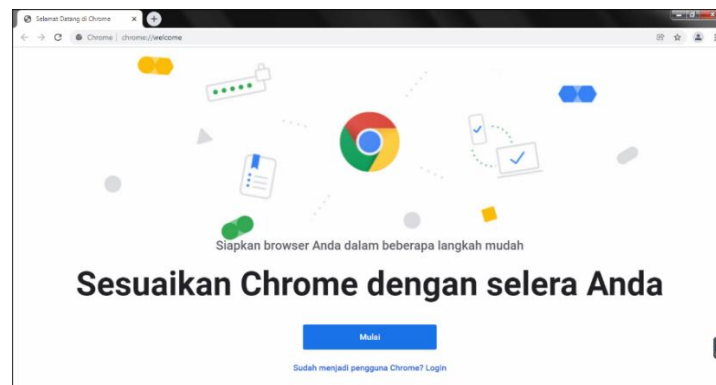
- 5) Selesai mengunduh, program secara otomatis akan menginstalnya, tunggu proses instalasi.



Gambar 3.12. Setup Memasang Googel Chrome

- 6) Jika aplikasi chrome sudah terbuka otomatis seperti ini berarti instalasi chrome telah berhasil dilakukan. *Browser* ini nantinya

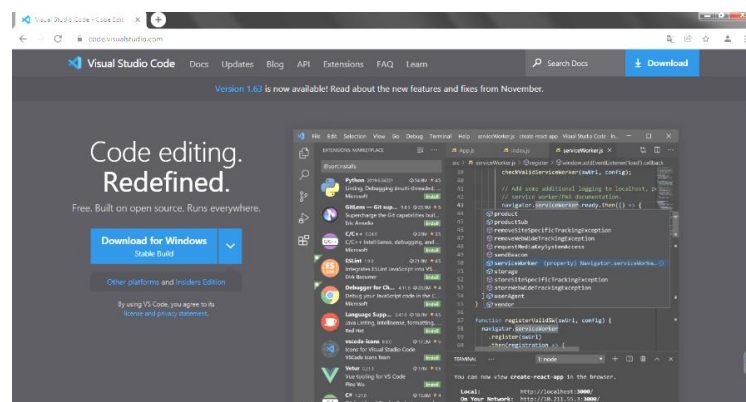
akan sering digunakan karena dalam pembuatan sebuah *website* memerlukan sebuah *browser*.



Gambar 3.13. Tampilan Aplikasi Google Chrome

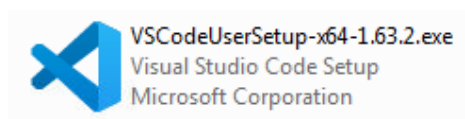
b. Instalasi Visual Studio Code

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



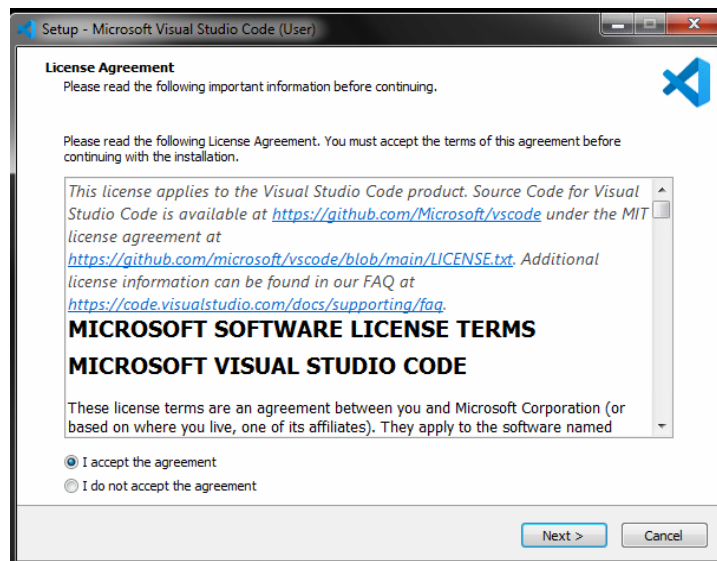
Gambar 3.14. Halaman *Website* Visual Studio Code

- 2) Unduh *file installer* visual studio, karena disini menggunakan windows maka pilih format installer untuk windows.
- 3) Jalankan file installer yang berhasil diunduh.



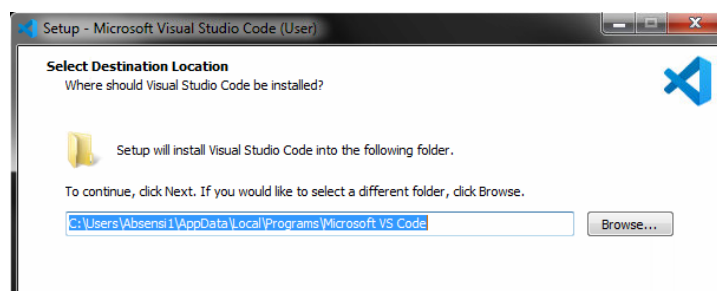
Gambar 3.15. *File Installer* Visual Studio Code

- 4) Pilih *I accept the agreement*, kemudian klik button next.



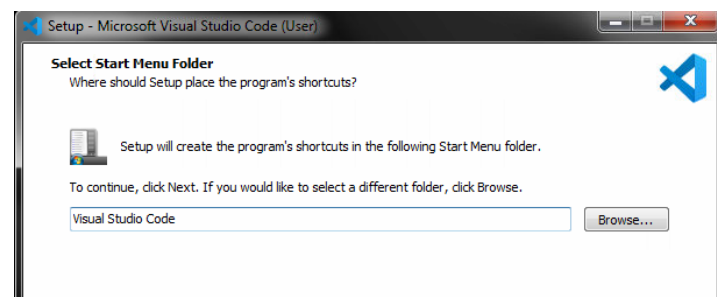
Gambar 3.16. Persetujuan Lisensi Visual Studio Code

- 5) Lantas pilih lokasi lokasi untuk aplikasi Visual Studio Code.



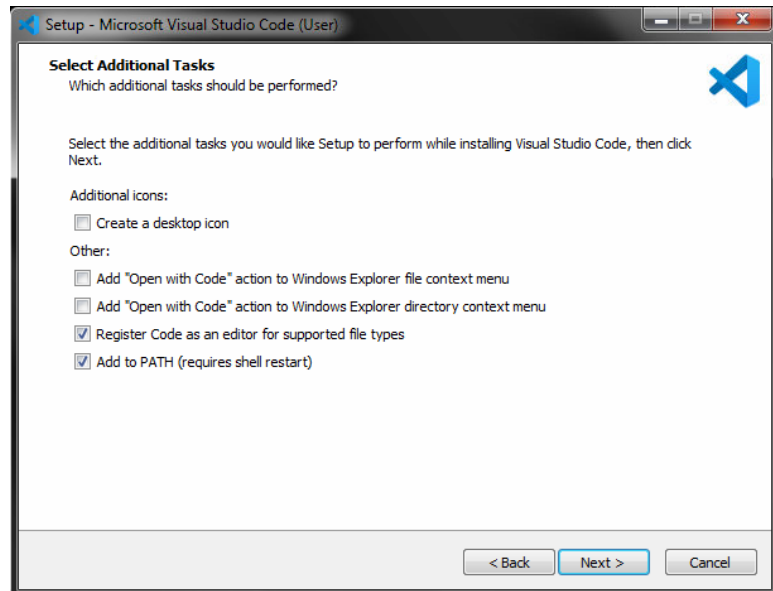
Gambar 3.17. Pemilihan Lokasi Visual Studio Code

- 6) Pilih lokasi untuk start menu folder, atau bisa klik next langsung jika ingin default.



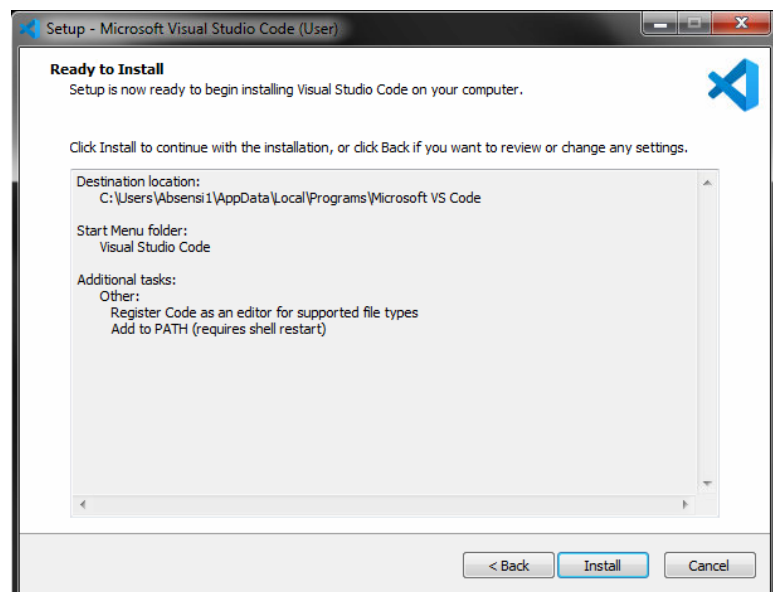
Gambar 3.18. Lokasi Start Menu Visual Studio Code

- 7) Kemudian pilih beberapa pilihan tambahan atau bisa langsung klik next jika ingin default.



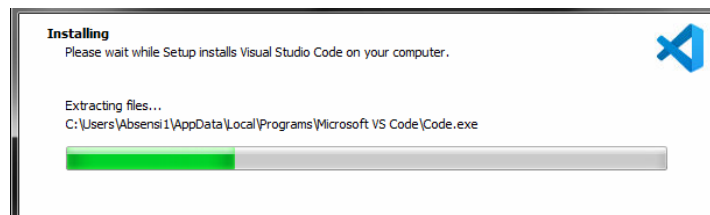
Gambar 3.19. Opsi Tambahan Instalasi Visual Studio Code

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



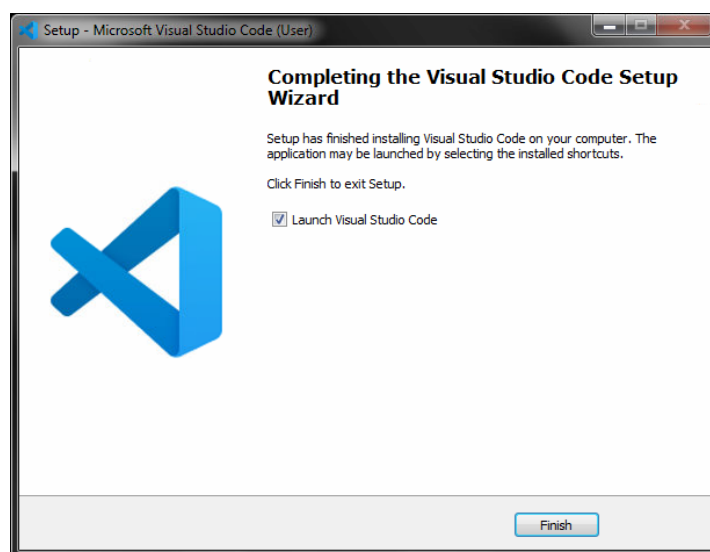
Gambar 3.20. Informasi Instalasi Visual Studio Code

9) Tunggu proses instalasi.



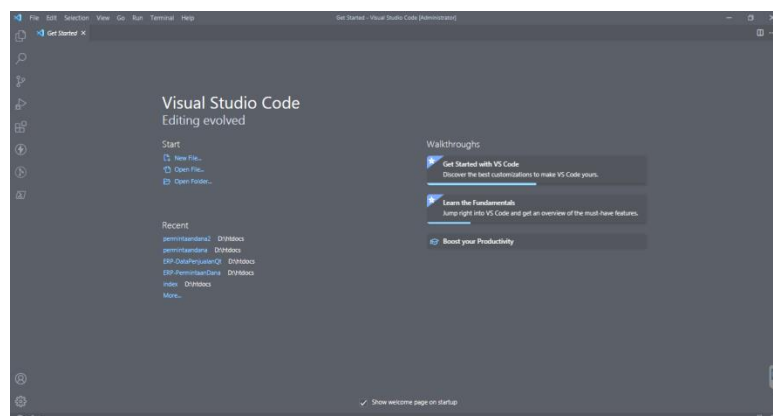
Gambar 3.21. Proses Instalasi Visual Studio Code

10) Klik tombol Finish.



Gambar 3.22. Instalasi Selesai Visual Studio Code

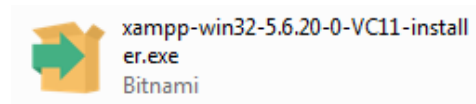
11) Jika Visual Studio Code sudah terbuka dan berjalan dengan normal, maka instalasi berhasil.



Gambar 3.23. Tampilan Aplikasi Visual Studio Code

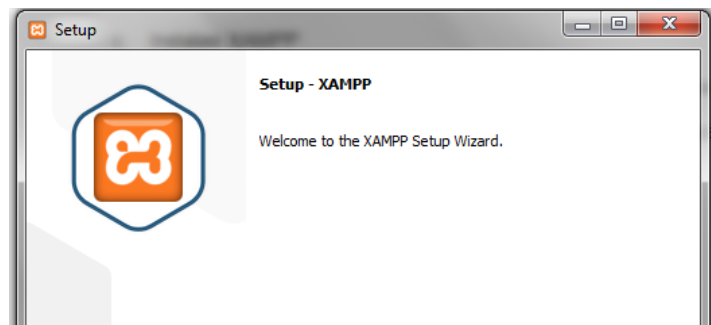
c. Instalasi XAMPP

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



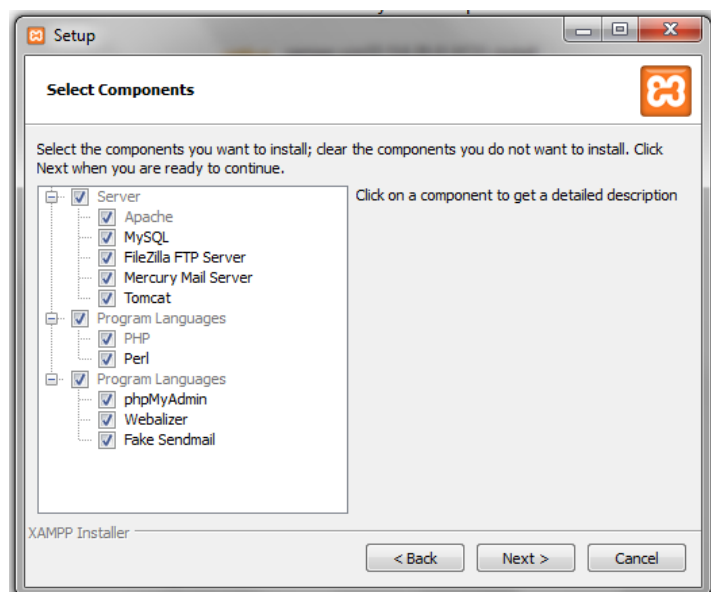
Gambar 3.24. File Installer XAMPP

- 2) Jalankan *file installer* tersebut, klik next.



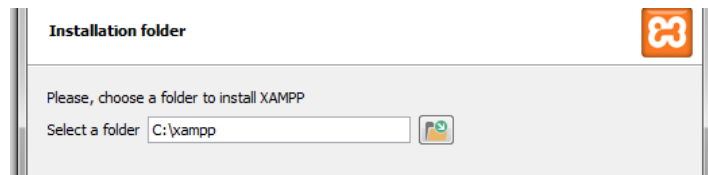
Gambar 3.25. Setup XAMPP

- 3) Klik next.



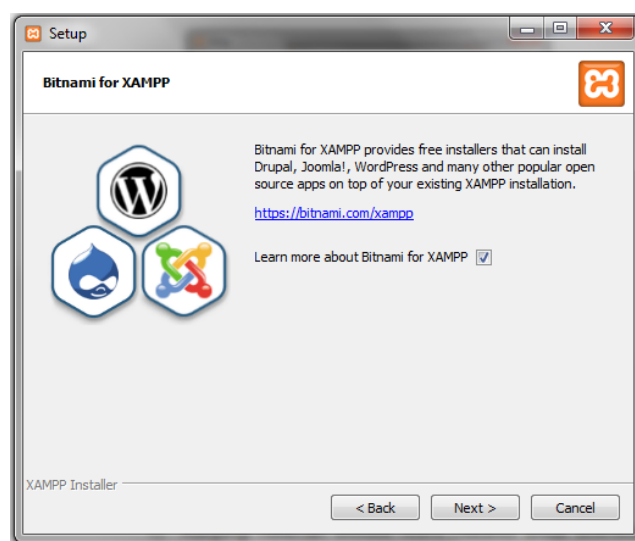
Gambar 3.26. Pilihan Komponen XAMPP

- 4) Pilih lokasi folder instalasi xampp, jika sudah klik next.



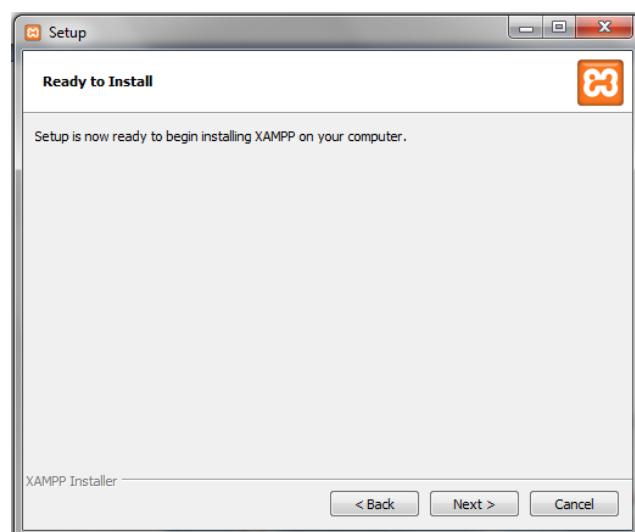
Gambar 3.27. Lokasi Folder XAMPP

- 5) Klik next.



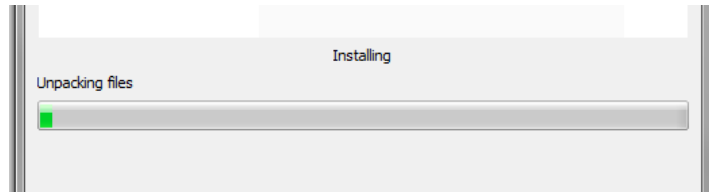
Gambar 3.28. Setup XAMPP

- 6) Klik next.



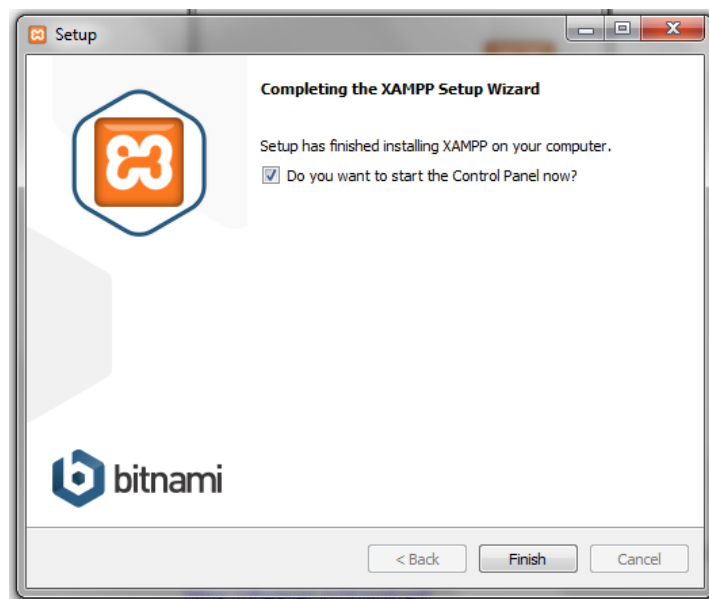
Gambar 3.29. Setup XAMPP

- 7) Tunggu proses instalasi hingga selesai.



Gambar 3.30. Proses Instalasi XAMPP

- 8) Kemudian klik finish, jika proses instalasi telah selesai.



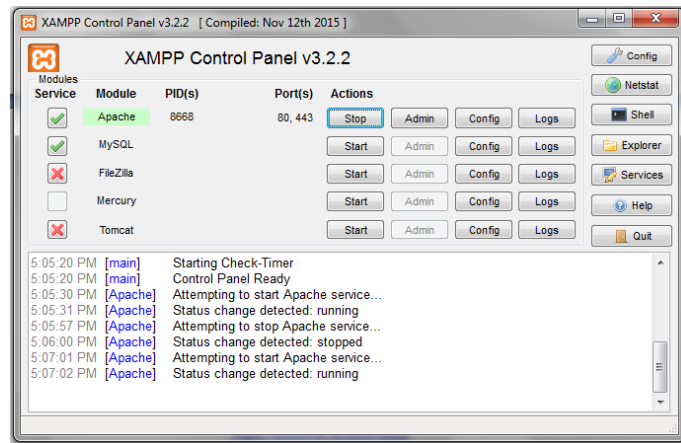
Gambar 3.31. Selesai Instalasi XAMPP

- 9) Pilih bahasa Inggris (bendera Amerika Serikat).



Gambar 3.32. Pemilihan Bahasa XAMPP

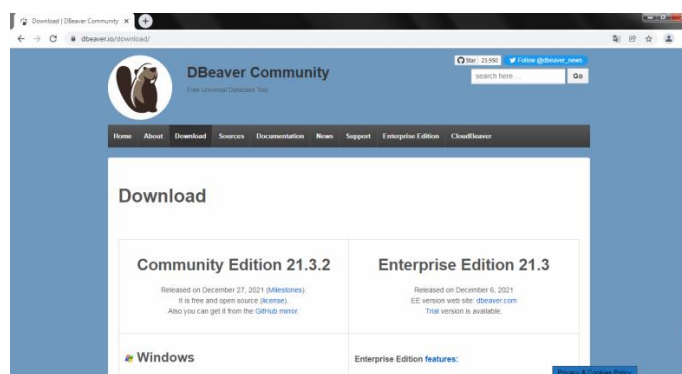
- 10) Coba jalankan service *Apache*, jika tidak ada masalah maka instalasi xampp untuk *web service* telah berhasil.



Gambar 3.33. Tampilan Aplikasi XAMPP

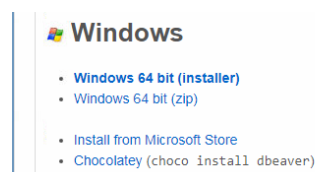
d. Instalasi DBeaver

- 1) Kunjungi halaman *website* resmi Dbeaver untuk download, <https://dbeaver.io/download/>.



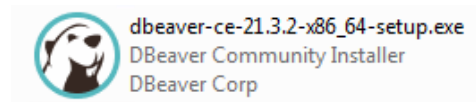
Gambar 3.34. Halaman Website DBeaver

- 2) Pilih DBeaver Community Edition dan sesuaikan dengan sistem operasi yang digunakan yaitu windows.



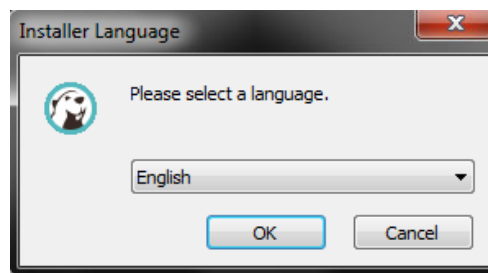
Gambar 3.35. Pemilihan Sistem Operasi DBeaver

- 3) Jalankan file installer yang berhasil diunduh tadi.



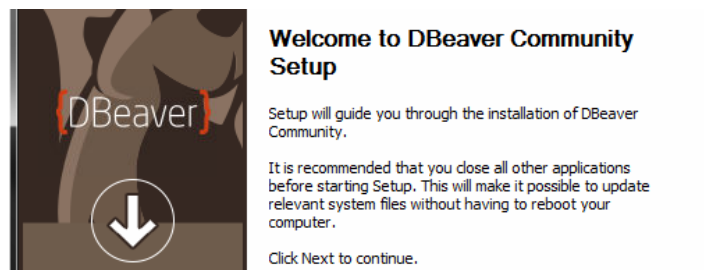
Gambar 3.36. *File Installer DBeaver*

- 4) Pilih bahas yang akan dipakai selama menginstall DBeaver.



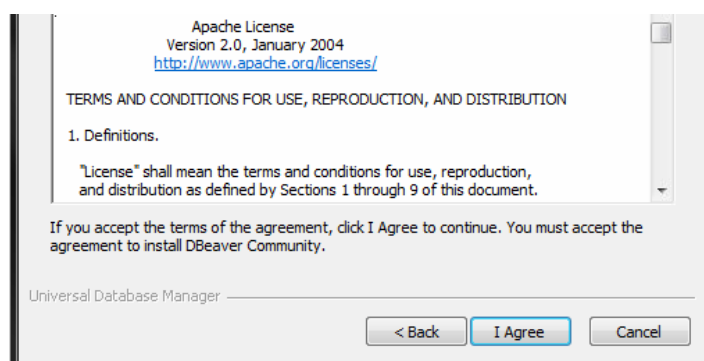
Gambar 3.37. Bahasa Instalasi DBeaver

- 5) Klik tombol next.



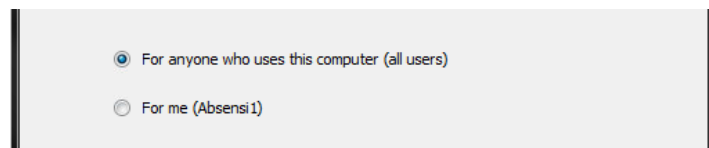
Gambar 3.38. Setup DBeaver

- 6) Klik tombol I Agree.



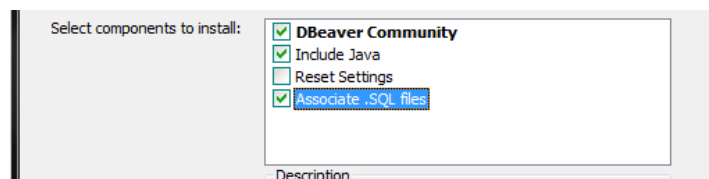
Gambar 3.39. Persetujuan Lisensi DBeaver

- 7) 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.



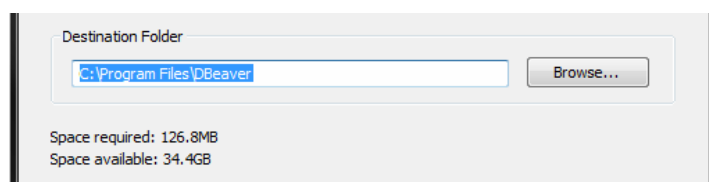
Gambar 3.40. User DBeaver

- 8) Pilih komponen apa saja yang akan diinstal, untuk pilihannya bisa disamakan dengan gambar.



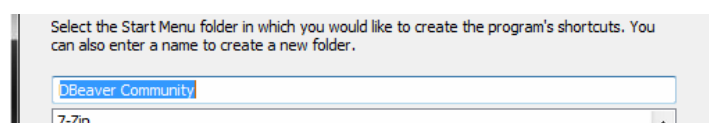
Gambar 3.41. Pilihan Komponen DBeaver

- 9) Pilih lokasi folder untuk DBeaver, jika ingin default langsung klik tombol next.



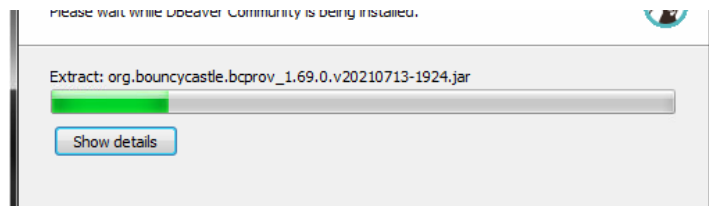
Gambar 3.42. Lokasi Folder DBeaver

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



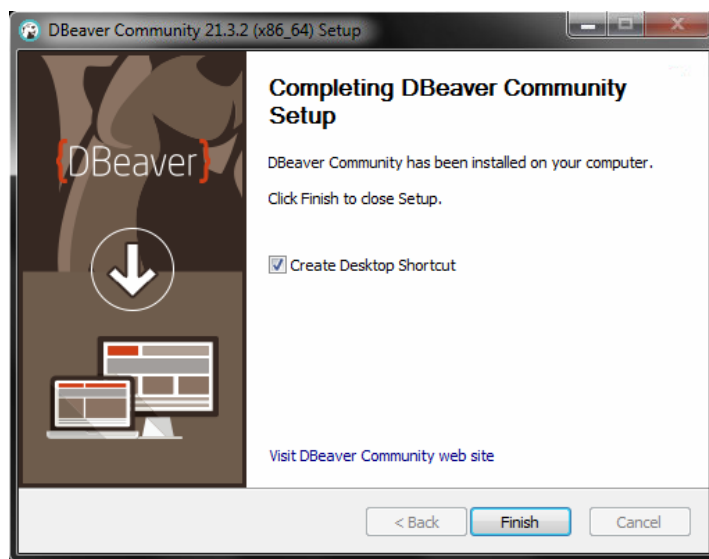
Gambar 3.43. Lokasi Folder Start Menu DBeaver

11) Tunggu proses instal DBeaver selesai.



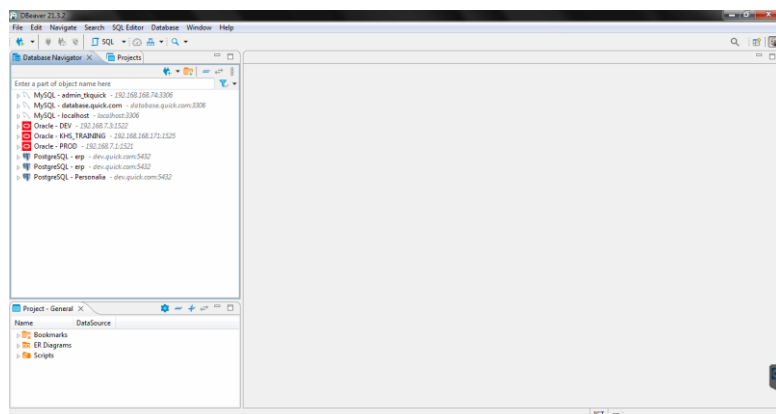
Gambar 3.44. Proses Instalasi DBeaver

12) Jika sudah selesai klik tombol finish.



Gambar 3.45. Selesai Instalasi DBeaver

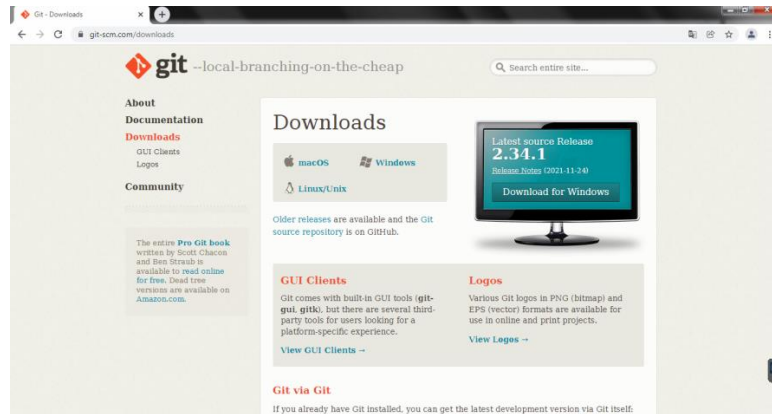
13) Kemudian jalankan aplikasi DBeaver, jika aplikasi terbuka dan normal, maka aplikasi berhasil diinstal.



Gambar 3.46. Tampilan Aplikasi DBeaver

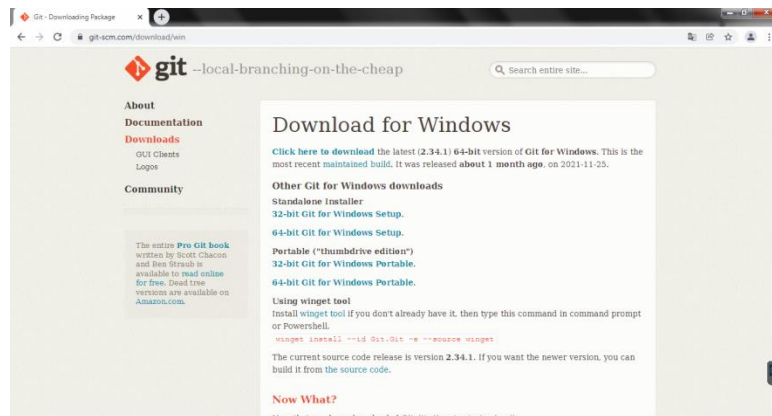
e. Instalasi Git

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



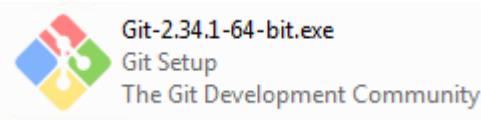
Gambar 3.47. Halaman *Website* Git

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



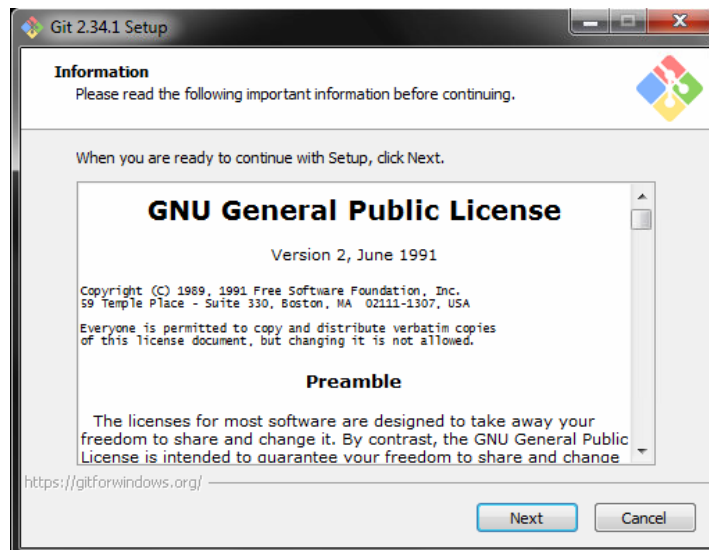
Gambar 3.48. Pemilihan Installer Git

- 4) Jalankan *file installer* yang berhasil diunduh tadi.



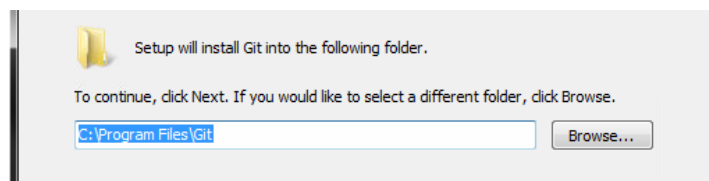
Gambar 3.49. *File Installer* Git

5) Klik tombol next.



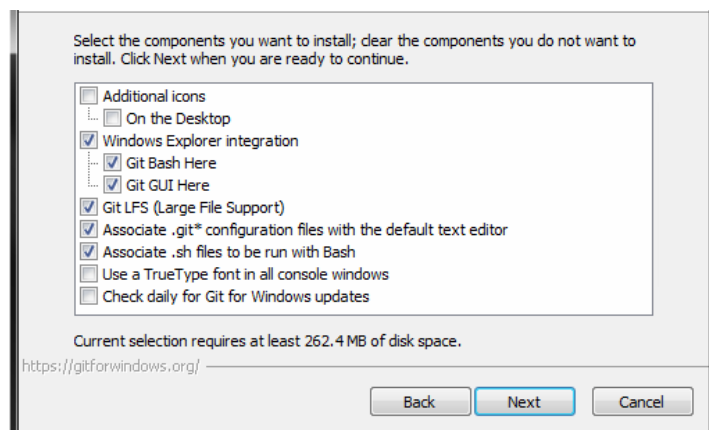
Gambar 3.50. Setup Git

6) Pilih lokasi aplikasi Git.



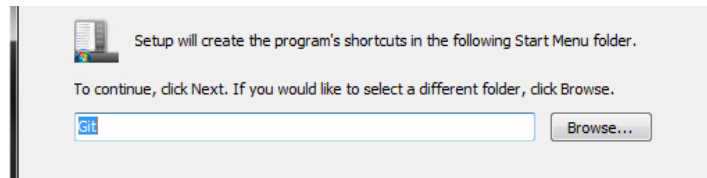
Gambar 3.51. Lokasi Folder Git

7) Pilih komponen apa saja yang akan diinstall, biarkan saja default langsung klik tombol next.



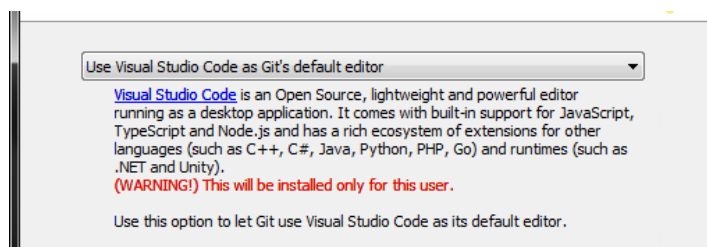
Gambar 3.52. Pilihan Komponen Git

- 8) Pilih lokasi folder git pada start menu, biarkan default langsung klik tombol next.



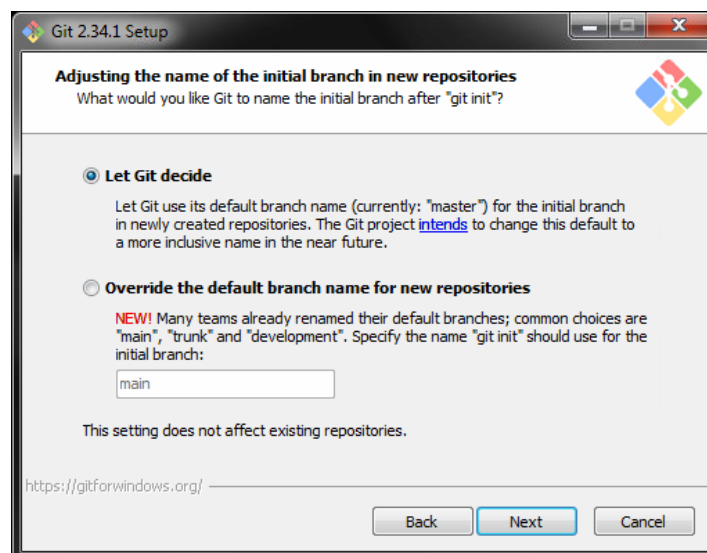
Gambar 3.53. Lokasi Folder Start Menu Git

- 9) Pilih *code editor* yang akan menggunakan git, karena disini menggunakan Visual Studi Code, maka pilih *Use Visual Studio Code as Git's default editor*.



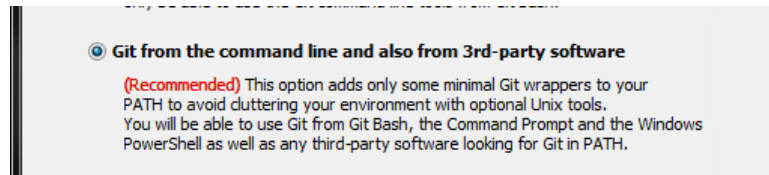
Gambar 3.54. Code Editor Untuk Git

- 10) Pilih *Let Git decide*.



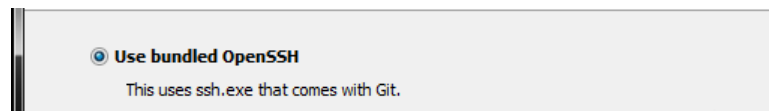
Gambar 3.55. Setup Git

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



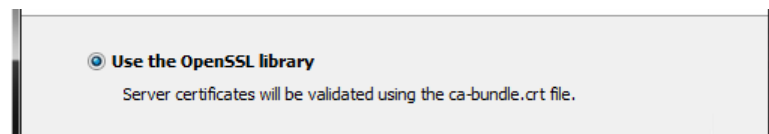
Gambar 3.56. Setup Git

- 12) Pilih *Use bundled OpenSSH*, klik next..



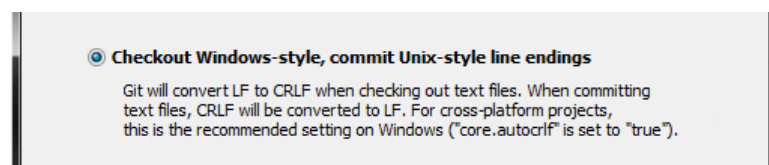
Gambar 3.57. Setup Git

- 13) Pilih *Use the OpenSSL library*, klik next..



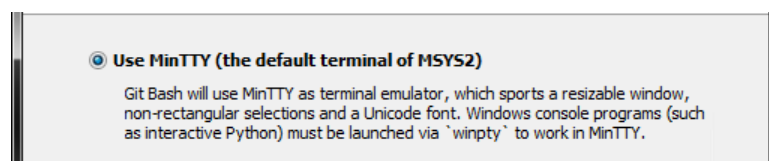
Gambar 3.58. Setup Git

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



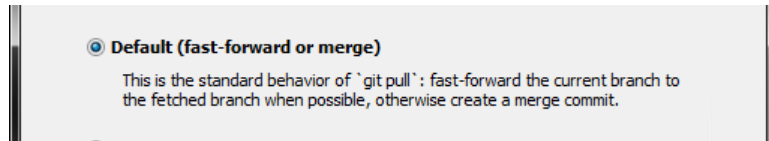
Gambar 3.59. Setup Git

- 15) Pilih *Use MinTTY*, klik next..



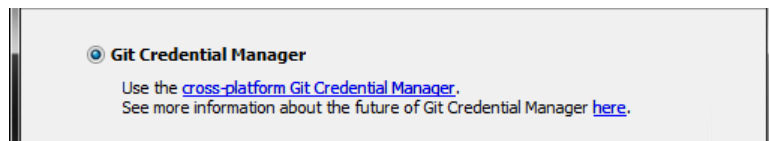
Gambar 3.60. Setup Git

16) Pilih *Default*, klik next.



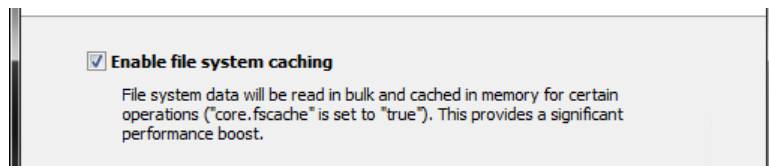
Gambar 3.61. Setup Git

17) Pilih *Git Credential Manager*, klik next.



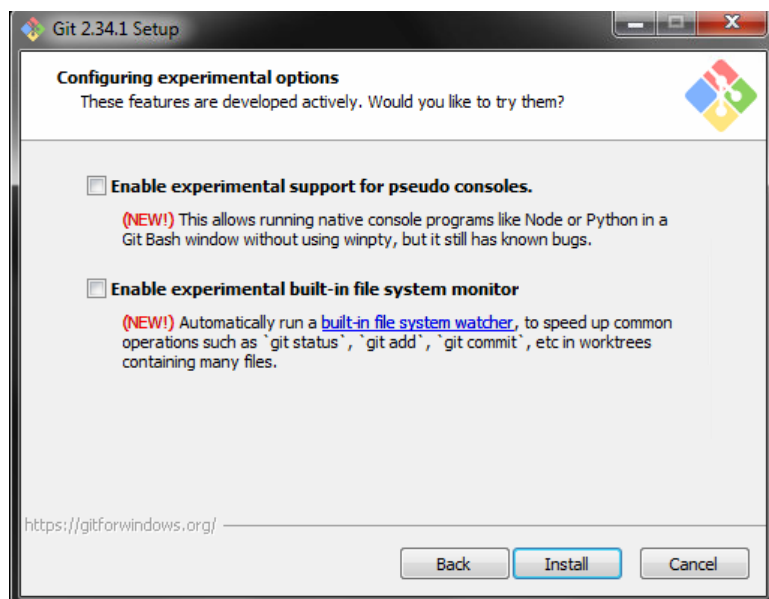
Gambar 3.62. Setup Git

18) Checklist *Enable file system caching*, klik next.



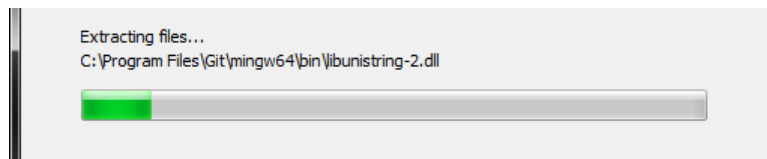
Gambar 3.63. Setup Git

19) Langsung klik tombol Install.



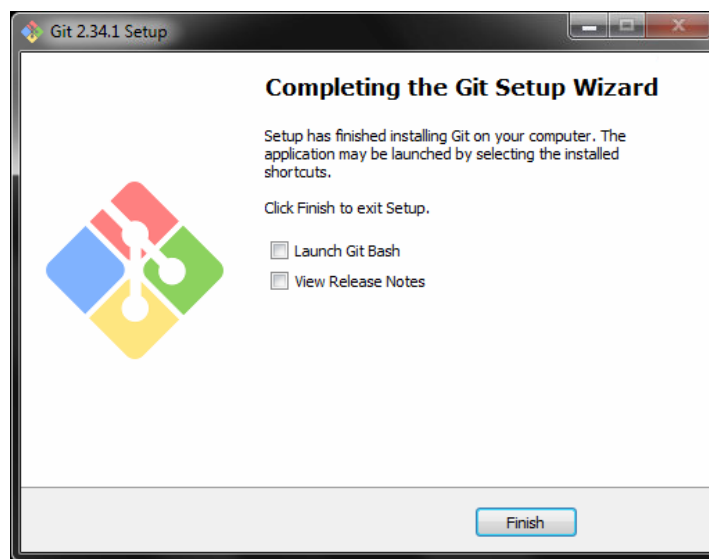
Gambar 3.64. Instal Git

20) Tunggu proses instalasi hingga selesai.



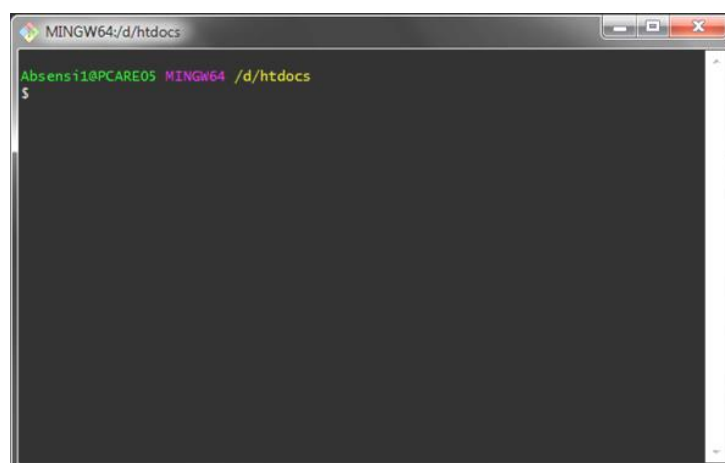
Gambar 3.65. Proses Instalasi Git

21) Jika sudah selesai klik tombol finish.



Gambar 3.66. Selesai Instalasi Git

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



Gambar 3.67. Tampilan Aplikasi Gitbash

f. Instalasi FileZilla

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



Gambar 3.68. Halaman *Website* FileZilla

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

Please select your edition of FileZilla Client

	FileZilla	FileZilla with manual	FileZilla Pro	FileZilla Pro + CLI
Standard FTP	Yes	Yes	Yes	Yes
FTP over TLS	Yes	Yes	Yes	Yes
SFTP	Yes	Yes	Yes	Yes
Comprehensive PDF manual	-	Yes	Yes	Yes
Amazon S3	-	-	Yes	Yes
Backblaze B2	-	-	Yes	Yes
Dropbox	-	-	Yes	Yes
Microsoft OneDrive	-	-	Yes	Yes
Google Drive	-	-	Yes	Yes
Google Cloud Storage	-	-	Yes	Yes
Microsoft Azure Blob + File Storage	-	-	Yes	Yes
WebDAV	-	-	Yes	Yes
OpenStack Swift	-	-	Yes	Yes
Box	-	-	Yes	Yes
Site Manager synchronization	-	-	Yes	Yes
Command-line interface	-	-	-	Yes
Batch transfers	-	-	-	Yes

Download
Select
Select
Select

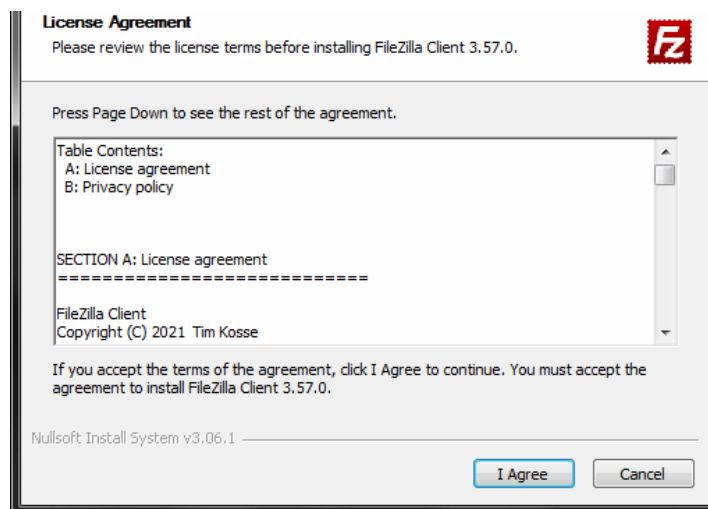
Gambar 3.69. Versi Aplikasi FileZilla

- 4) Jalankan *file installer* yang telah berhasil diunduh tadi.



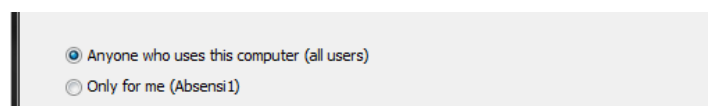
Gambar 3.70. *File Installer* FileZilla

- 5) Klik tombol I Agree.



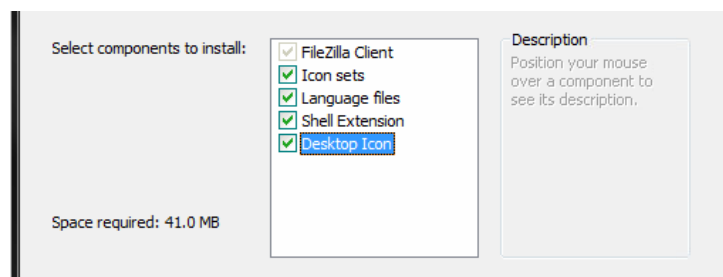
Gambar 3.71. Persetujuan Lisensi FileZilla

- 6) Pilih *Anyone who uses this computer*, karena ini bukan komputer pribadi.



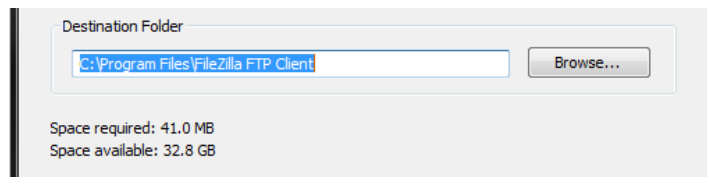
Gambar 3.72. User FileZilla

- 7) Untuk komponen yang akan diinstall pilih semuanya.



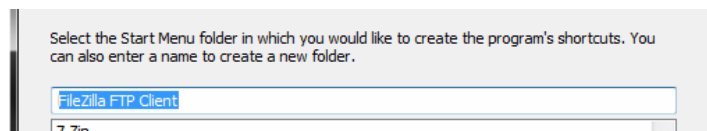
Gambar 3.73. Pilihan Komponen FileZilla

- 8) Untuk lokasi aplikasi FileZilla biarkan default.



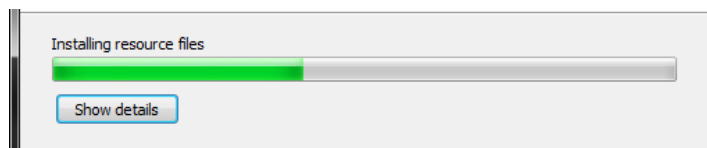
Gambar 3.74. Lokasi Folder FileZilla

- 9) Folder start menu untuk aplikasi FileZilla biarkan default, langsung klik tombol Install.



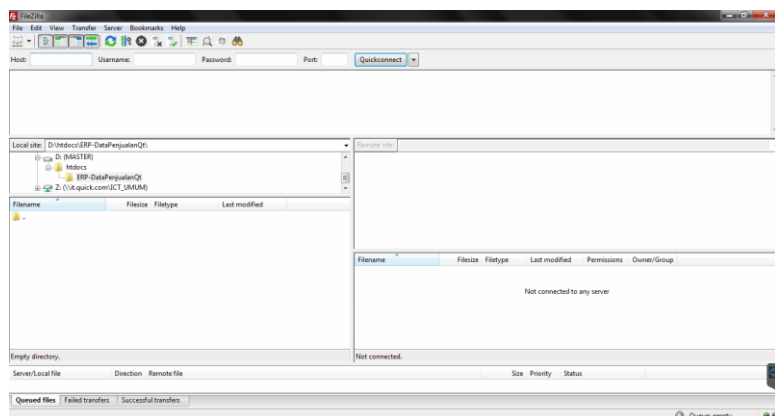
Gambar 3.75. Lokasi Folder Start Menu FileZilla

- 10) Tunggu proses instalasi hingga selesai.



Gambar 3.76. Proses Instalasi FileZilla

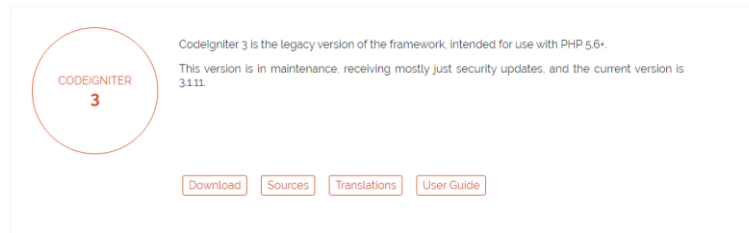
- 11) Kemudian klik tombol Finish.
- 12) Jika aplikasi FileZilla sudah berjalan dengan baik, maka instalasi telah berhasil dilakukan.



Gambar 3.77. Tampilan Aplikasi FileZilla

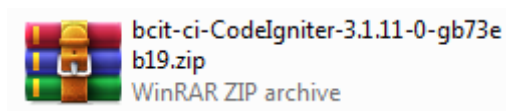
g. Instalasi Codeigniter

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



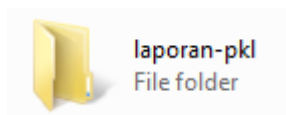
Gambar 3.78. Codeigniter 3

- 3) Hasil download berupa file dengan format zip.



Gambar 3.79. File Setup Codeigniter 3

- 4) Buat sebuah folder pada *directoryweb server*.



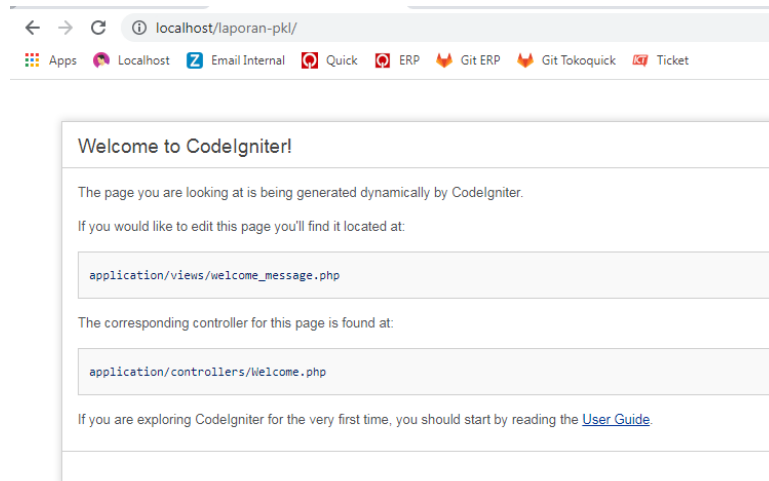
Gambar 3.80. Folder Penampung Codeigniter

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

application	9/19/2019 7:08 PM	File folder	
system	9/19/2019 7:08 PM	File folder	
user_guide	9/19/2019 7:08 PM	File folder	
.editorconfig	9/19/2019 7:08 PM	Editor Config Sour...	1 KB
.gitignore	9/19/2019 7:08 PM	Text Document	1 KB
composer.json	9/19/2019 7:08 PM	JSON File	1 KB
contributing.md	9/19/2019 7:08 PM	MD File	7 KB
index.php	9/19/2019 7:08 PM	PHP File	11 KB
license.txt	9/19/2019 7:08 PM	Text Document	2 KB
readme.rst	9/19/2019 7:08 PM	Restructured Text ...	3 KB

Gambar 3.81. Struktur File Codeigniter

- 6) Untuk memastikan berhasil atau tidaknya, buka browser dan buka alamat “localhost/nama_folder”.



Gambar 3.82. Tampilan *Website* Codeigniter

7. 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.

8. Kesimpulan

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

B. Produk PKL 2 : Penggunaan Git di Lingkup Perusahaan

1. 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 terjalannya kerja sama untuk melakukan kolaborasi bersama.

Salah satunya adalah Git, sebuah *version control system* yang digunakan para *programmer* untuk mengembangkan *software* secara bersama-sama. 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.

2. Alat dan Bahan yang Digunakan

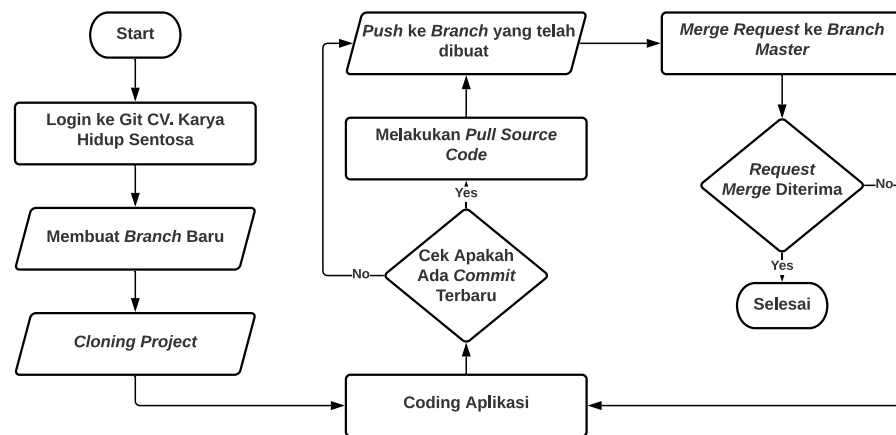
- a. Komputer dan *peripheral*-nya
- b. Sistem Operasi Windows 7
- c. Visual Studio Code
- d. *Browser* Chrome
- e. Gitbash

3. Keselamatan Kerja

- a. Niat dan berdoa sebelum melakukan kegiatan.
- b. Memastikan semua alat dan bahan yang diperlukan sudah tersedia.
- c. Menggunakan peralatan sesuai fungsinya.
- d. Gunakan internet sebagaimana mestinya.
- e. Posisi duduk tegap.
- f. Memperhatikan jarak pandang mata dengan monitor minimal 30 cm.

- g. Kurangi kecerahan monitor supaya nyaman untuk mata.
- h. Hindari makan dan minum di dekat monitor.
- i. Matikan perangkat bila sudah tidak digunakan.
- j. Bersihkan perangkat yang digunakan bila kotor.

4. Gambar Kerja



Gambar 3.83. Gambar Kerja Penggunaan Git

5. Analisis

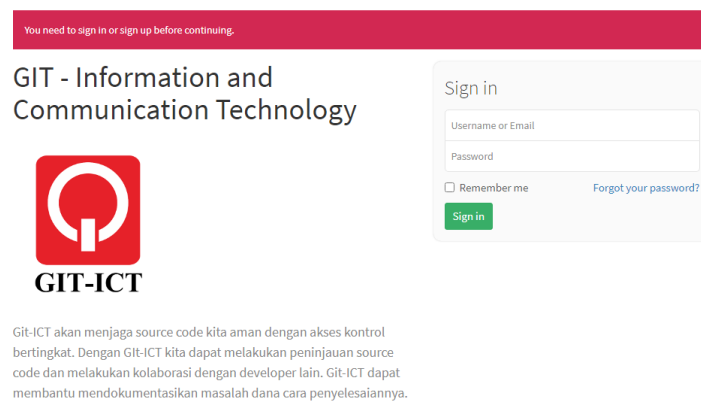
Didalam pembuatan dan pengembangan aplikasi baik web maupun android pastinya dibuat tim supaya pembuatan dan pengembangan aplikasi lebih cepat dan tidak memakan banyak waktu. Seorang *programmer* pastinya ada beberapa bahasa pemrograman yang tidak mereka kuasai, beberapa ada yang tidak ahli dalam mendesain 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* mereka, sehingga program menjadilebih terstruktur, tertata, dan mudah dipahami sesama *programmer*.

6. Langkah Kerja

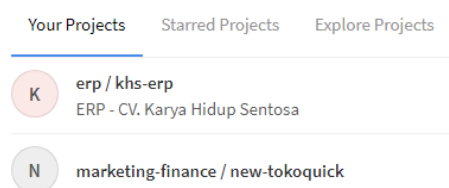
a. Login ke Git Quick

- 1) Buka halama Git Quick menggunakan *browser*. *Login* dengan menggunakan username dan password yang telah didaftarkan oleh atasan.



Gambar 3.84. Halaman Login Git Quick

- 2) Maka akan muncul daftar project, daftar ini berdasarkan project yang diikuti oleh user.

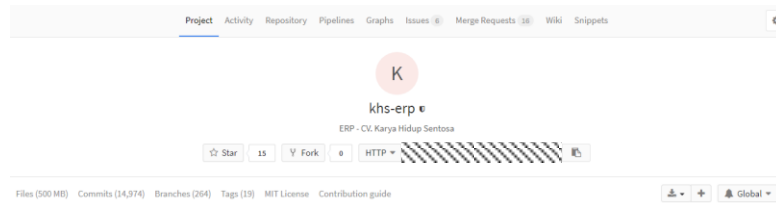


Gambar 3.85. Daftar Project Git

b. Membuat branch baru di Git Quick

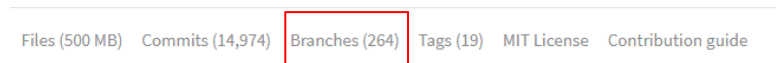
- 1) Buka salah satu project. Disini penulis membuka *projecterp/khs-erp*.

- 2) Anda akan memasuki halaman awal *project erp/khs-erp*.



Gambar 3.86. Halaman Git Quick

- 3) Kemudian klik tab *Branches*.



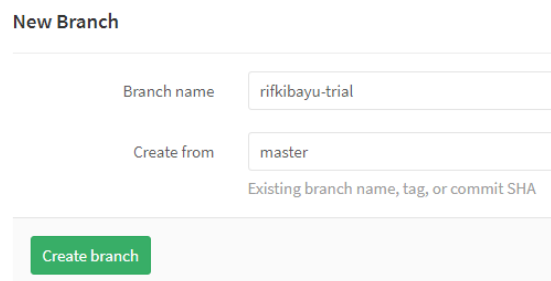
Gambar 3.87. Tab Pada Menu Project

- 4) Klik *New Branch* untuk beralih ke halaman yang berfungsi untuk membuat *branch* baru.



Gambar 3.88. Tombol New *branch*

- 5) 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.

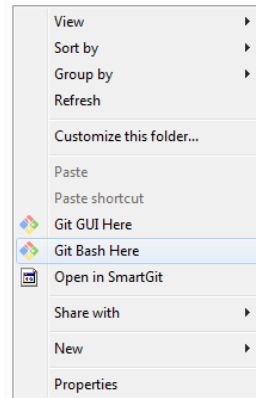


Gambar 3.89. *Form* Buat *Branch* Baru

- 6) Jika sudah klik button *Create Branch* untuk memproses pembuatan *branch* baru.

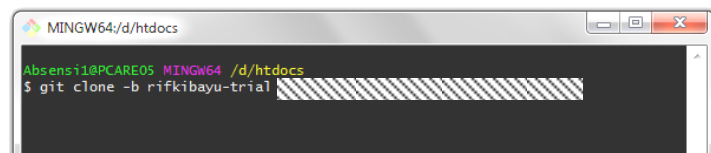
c. Clone branch ke directory local

- 1) Buka directory *web server* local menggunakan *file explorer*.
- 2) Buka *gitbash* yang mengarah ke *directory* tersebut, dengan cara klik kanan pada *directory* tersebut.



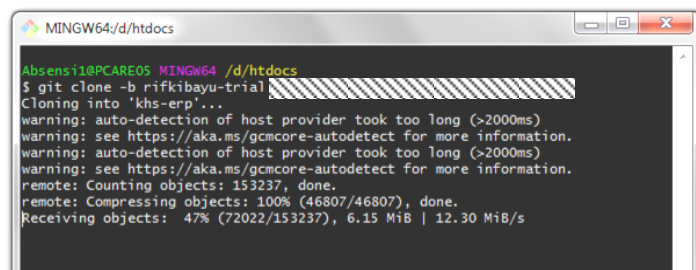
Gambar 3.90. Pilihan Membuka Gitbash Pada Direktori

- 3) Kemudian jalankan baris perintah seperti berikut.



Gambar 3.91. Code Untuk CloneBranch

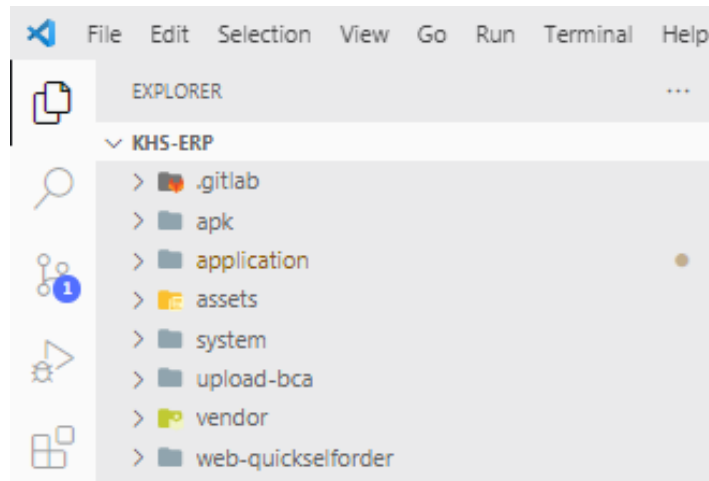
- 4) Script yang dijalankan, `git clone -b nama_branch link_repository`, tekan enter untuk menjalankannya.
- 5) Kemudian tunggu proses *clone branch* hingga berhasil.



Gambar 3.92. Proses CloneBranch

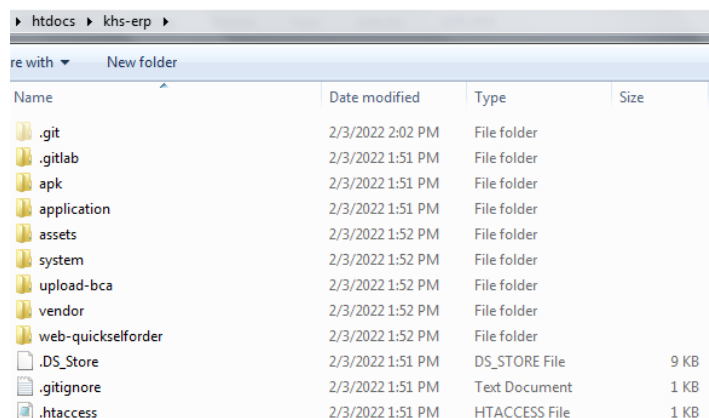
d. Push *commit* ke branch

- 1) Buka folder *repository* yang berhasil diclone tadi melalui Visual Studio Code.



Gambar 3.93. Project Pada Visual Studio Code

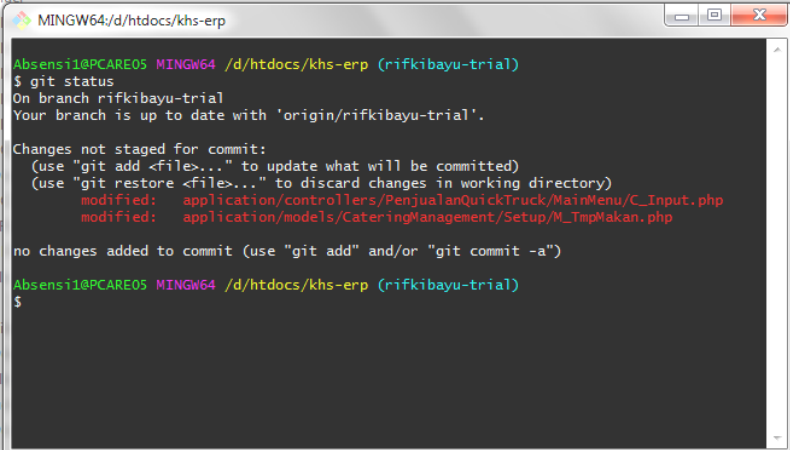
- 2) Coba ubah salah satu file yang berada pada *directory* tersebut menggunakan *code editor*.
- 3) Jika sudah berhasil diubah, buka *directory* hasil clone tadi menggunakan *file explorer*.



Gambar 3.94. Project Hasil Clone

- 4) Lantas buka gitbash yang mengarah ke *directory* tersebut.

- 5) Jalankan perintah *git status*, untuk mengecek perubahan apa saja yang telah terjadi.



```

Absensi1@PCARE05 MINGW64 /d/htdocs/khs-erp (rifkibayu-trial)
$ git status
On branch rifkibayu-trial
Your branch is up to date with 'origin/rifkibayu-trial'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   application/controllers/PenjualanQuickTruck/MainMenu/C_Input.php
        modified:   application/models/CateringManagement/Setup/M_TmpMakan.php

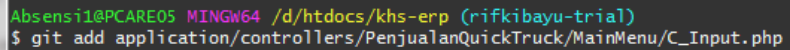
no changes added to commit (use "git add" and/or "git commit -a")

Absensi1@PCARE05 MINGW64 /d/htdocs/khs-erp (rifkibayu-trial)
$

```

Gambar 3.95. Perubahan Pada *Branch*

- 6) Terdapat dua file yang termodifikasi.
- 7) Disini penulis hanya akan commit salah satu file saja. Menggunakan perintah *git add lokasi_file_modifikasi*.



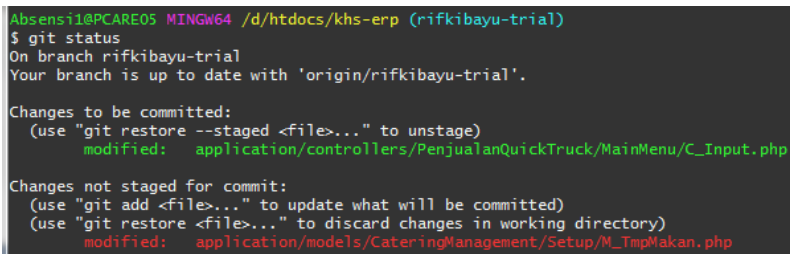
```

Absensi1@PCARE05 MINGW64 /d/htdocs/khs-erp (rifkibayu-trial)
$ git add application/controllers/PenjualanQuickTruck/MainMenu/C_Input.php

```

Gambar 3.96. Menambahkan File *Commit* Git

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



```

Absensi1@PCARE05 MINGW64 /d/htdocs/khs-erp (rifkibayu-trial)
$ git status
On branch rifkibayu-trial
Your branch is up to date with 'origin/rifkibayu-trial'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   application/controllers/PenjualanQuickTruck/MainMenu/C_Input.php

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   application/models/CateringManagement/Setup/M_TmpMakan.php

```

Gambar 3.97. Status *CommitBranch*

- 9) Kemudian tambahkan keterangan *commit* dengan perintah *git commit -m "keterangan"*.

```
Absensi1@PCARE05 MINGW64 /d/htdocs/khs-erp (rifkibayu-trial)
$ git commit -m "trial push"
[rifkibayu-trial 676be9e37] trial push
1 file changed, 12 insertions(+), 1 deletion(-)
```

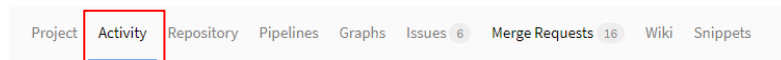
Gambar 3.98. Keterangan *Commit*

- 10) Jika sudah, langkah terakhir jalankan perintah *git push origin* untuk push ke branch.

```
Absensi1@PCARE05 MINGW64 /d/htdocs/khs-erp (rifkibayu-trial)
$ git push origin
warning: auto-detection of host provider took too long (>2000ms)
warning: see https://aka.ms/gcmcore-autodetect for more information.
warning: auto-detection of host provider took too long (>2000ms)
warning: see https://aka.ms/gcmcore-autodetect for more information.
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 709 bytes | 709.00 KiB/s, done.
Total 7 (delta 5), reused 0 (delta 0), pack-reused 0
```

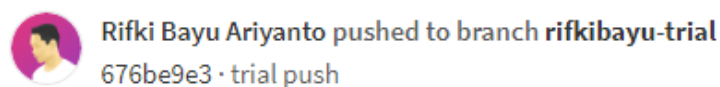
Gambar 3.99. Push *Commit* Ke *Branch*

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



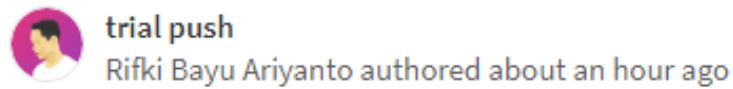
Gambar 3.100. Menu Git Quick

- 3) Cari activity push tadi, jika ketemu klik pada nama *branch*.



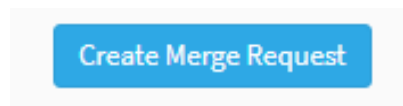
Gambar 3.101. History Git Quick

- 4) Pilih *commit* yang sesuai dengan keterangan pada saat push *commit* tadi.



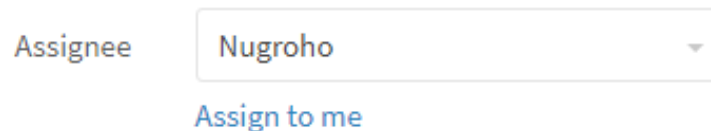
Gambar 3.102. *Commit Pada Branch*

- 5) Cek kembali file file yang telah dimodifikasi tadi.
- 6) Kembali ke *tab activity*, klik button *Create Merge Request* di pojok atas kanan.



Gambar 3.103. Tombol *Merge Request*

- 7) Kemudian pilih nama atasan yang akan menyetujui request merge.



Gambar 3.104. Target Approve *Merge*

- 8) Jika sudah klik tombol *Submit merge request* untuk meminta persetujuan merge ke atasan yang dipilih tadi.

 A screenshot of a GitHub "Form Merge Request" form. It features two dropdown menus: "Source branch" with the value "rifkibayu-trial" and "Target branch" with the value "master". To the right of the "Target branch" dropdown is a link that says "Change branches". Below these dropdowns is a checkbox with the label "Remove source branch when merge request is accepted." At the bottom of the form is a green button labeled "Submit merge request". To the right of the button, there is a text prompt: "Please review the [contribution guidelines](#) for this project."

Gambar 3.105. *FormMerge Request*

- 9) Tunggu diapprove oleh atasan.

7. 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, pengecekan *source code*, dan develop aplikasi tersebut ditunda terlebih dahulu karena suatu hal yang lebih urgent.

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

- a. 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 project tersebut termasuk project *erp* maka *merge* ke project *erp*.
- b. 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.
- c. Jika atasan memerintahkan kita untuk melakukan pengecekan *source code*, biasanya atasan menyuruh *programmer* senior untuk melakukan pengecekan *source code*. Akan tetapi sebelum menyerahkan ke senior, kita coba cek sendiri terlebih dahulu.

- d. Jika atasan memerintahkan kita untuk menunda develop aplikasi yang dikerjakan, maka kita berhenti untuk develop aplikasi tersebut. Lalu menunggu perintah atasan selanjutnya.

8. 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.

C. Produk PKL 3 : Pembuatan Aplikasi Data Penjualan Quick Truck

1. Penjelasan Singkat Jenis Pekerjaan atau Produk

Aplikasi Data Penjualan Quick Truck merupakan aplikasi yang digunakan untuk mengurus 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.

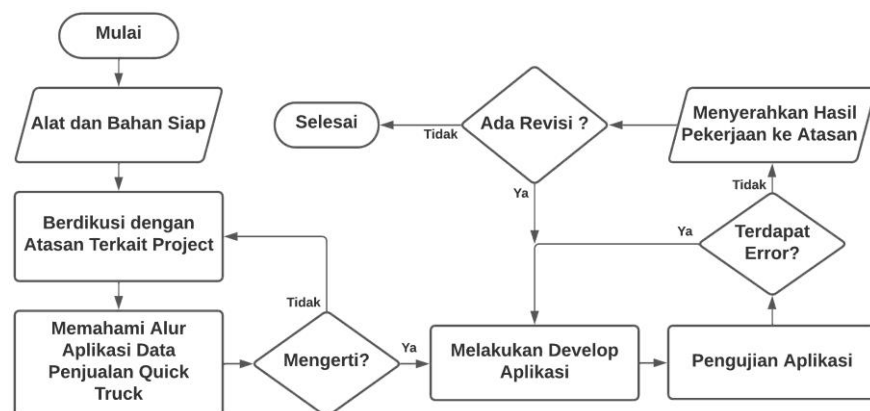
2. Alat dan Bahan yang Digunakan

- Komputer dan *peripheral*-nya
- Sistem Operasi Windows 7
- Visual Studio Code
- Browser* Chrome

3. Keselamatan Kerja

- Niat dan berdoa sebelum melakukan kegiatan.
- Memastikan alat dan bahan yang diperlukan sudah tersedia.
- Menggunakan peralatan sesuai fungsinya.
- Gunakan internet sebagaimana mestinya.
- Posisi duduk tegap.
- Memperhatikan jarak pandang mata ke monitor minimal 30 cm.
- Kurangi kecerahan monitor supaya nyaman untuk mata.
- Hindari makan dan minum di dekat monitor.
- Matikan perangkat bila sudah tidak digunakan.
- Bersihkan perangkat yang digunakan bila kotor.

4. Gambar Kerja



Gambar 3.106. Gambar Kerja Aplikasi Data Penjualan Quick Truck

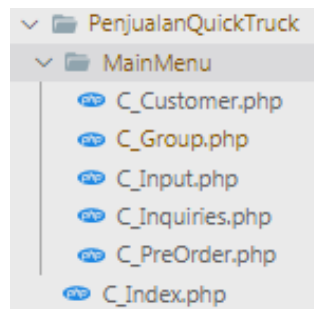
5. Analisis

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.

6. Langkah Kerja

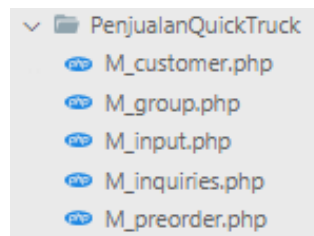
- a. Setelah sebelumnya telah melakukan *clone branch* dari remote *repository* Quick ERP pada Git CV. Karya Hidup Sentosa, langkah pertama yang dilakukan adalah mengganti koneksi ke *Database* Quick Dev. *Database* Quick Dev ini dikhususkan untuk *programmer* dalam melakukan develop aplikasi, sebelum nantinya dipublikasikan dan digunakan oleh user.
- b. 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. Sedangkan *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*:



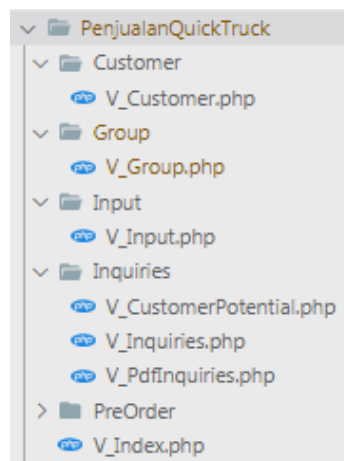
Gambar 3.107. Struktur *Controller* Aplikasi Quick Truck

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

- c. 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:

```
// ----- Penjualan Quick Truck -----
$route['PenjualanQuickTruck'] = 'C_Index';

$route['PenjualanQuickTruck/Group'] = 'C_Group';
$route['PenjualanQuickTruck/Group/(:any)'] = 'C_Group/$1';

$route['PenjualanQuickTruck/Customer'] = 'C_Customer';
$route['PenjualanQuickTruck/Customer/(:any)'] = 'C_Customer/$1';

$route['PenjualanQuickTruck/ inputData'] = 'C_Input';
$route['PenjualanQuickTruck/ inputData/(:any)'] = 'C_Input/$1';

$route['PenjualanQuickTruck/DataInquiries'] = 'C_Inquiries';
$route['PenjualanQuickTruck/DataInquiries/(:any)'] = 'PenjualanQuickTruck/MainMenu/C_Inquiries/$1';
$route['PenjualanQuickTruck/DataInquiries/(:any)/(:any)'] = 'C_Inquiries/$1/$2';

$route['PenjualanQuickTruck/CustomerPotential'] = 'C_Inquiries/CustomerPotential';
$route['PenjualanQuickTruck/CustomerPotential/(:any)'] = 'C_Inquiries/$1';

$route['PenjualanQuickTruck/PreOrder'] = 'C_PreOrder';
$route['PenjualanQuickTruck/PreOrder/(:any)'] = 'C_PreOrder/$1';

$route['PenjualanQuickTruck/DataSeries'] = 'C_PreOrder/DataSeries';
$route['PenjualanQuickTruck/DataSeries/(:any)'] = 'C_PreOrder/$1';
```

Gambar 3.110. *Routes* Aplikasi Quick Truck

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

```
function InstalasiClass() { ...
}

function ClearInput(element) { ...
}

function InputRequired(element) { ...
}

function SwalInputRequired(col) { ...
}

function AjaxSaveForm(link, formdata) { ...
}

function ModalBody(link, id, element) { ...
}

function AjaxDelete(name, link, id) { ...
}

function CheckTypeCustomer() { ...
}

function CheckCustomerSelectId(element) { ...
}

function CheckCustomerId() { ...
}

function AjaxExportReport(url, date) { ...
}
```

Gambar 3.111. *Function JavaScript* Aplikasi Quick Truck

- f. Membuat menu Master Group, *routes* yang dipakai:

```
$route['PenjualanQuickTruck/Group'] = 'C_Group';
$route['PenjualanQuickTruck/Group/(:any)'] = 'C_Group/$1';
```

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.

```
public function index()
{
    $user_id = $this->session->userid;

    $data['Menu'] = 'Data Penjualan Quick Truck';
    $data['SubMenuOne'] = 'Master Group QT';
    $data['Title'] = 'Master Group';
    $data['UserMenu'] = $this->M_user->getUserMenu($user_id, $this->session->responsibility_id);
    $data['UserSubMenuOne'] = $this->M_user->getMenuLv2($user_id, $this->session->responsibility_id);
    $data['UserSubMenuTwo'] = $this->M_user->getMenuLv3($user_id, $this->session->responsibility_id);
    $data['hak_akses'] = $this->UserAccess()['HAK_AKSES'];

    $this->load->view('V_Header', $data);
    $this->load->view('V_Sidemenu', $data);
    $this->load->view('PenjualanQuickTruck/Group/V_Group', $data);
    $this->load->view('V_Footer', $data);
}
```

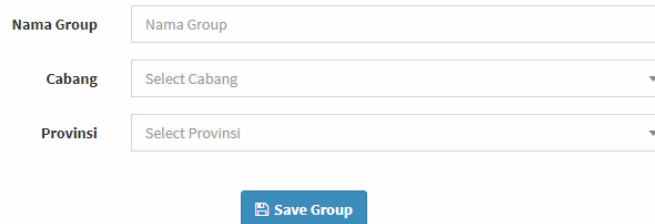
Gambar 3.113. Kode Memanggil *View* Menu Group

- 3) Langkah selanjutnya buka file V_Group untuk membuat tampilan yang berisikan kolom input group.

```
<form class="dpqt-group-formsave form-horizontal">
    <div class="form-group">
        <label class="col-sm-3 control-label">Nama Group</label>
        <div class="col-sm-9">
            <input type="text" name="group_name" autocomplete="off"
                class="form-control input-required" placeholder="Nama Group">
        </div>
    </div>
    <div class="form-group">
        <label class="col-sm-3 control-label">Cabang</label>
        <div class="col-sm-9">
            <select name="org_id" data-placeholder="Select Cabang"
                class="form-control dpqt-select-branch input-required">
                <option value=""></option>
            </select>
        </div>
    </div>
    <div class="form-group">
        <label class="col-sm-3 control-label">Provinsi</label>
        <div class="col-sm-9">
            <select name="province" data-placeholder="Select Provinsi"
                class="form-control dpqt-select-province input-required">
                <option value=""></option>
            </select>
        </div>
    </div>
    <br>
    <div class="form-group">
        <div class="col-sm-12" style="display:flex">
            <button class="btn btn-primary m-auto">
                <b><i class="fa fa-save mr-2"></i>Save Group</b>
            </button>
        </div>
    </div>
</form>
```

Gambar 3.14. Kode *Html View* Menu Group

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



The form consists of three input fields stacked vertically. The first is a text input labeled 'Nama Group'. The second is a dropdown menu labeled 'Cabang' with the placeholder text 'Select Cabang'. The third is a dropdown menu labeled 'Provinsi' with the placeholder text 'Select Provinsi'. Below these fields is a blue button with a white document icon and the text 'Save Group'.

Gambar 3.115. Tampilan View Menu Group

- 5) Kembali ke file *controller* C_Group, buat function SaveGroup yang nantinya akan digunakan untuk menangkap data dari View dan menyimpan ke *database* melalui *Model*.

```
public function SaveGroup()
{
    $this->M_group->SaveGroup(
        $_POST['group_name'],
        $_POST['org_id'],
        $_POST['province'],
        $this->session->user
    );
    echo json_encode(1);
}
```

Gambar 3.116. Function Menyimpan Group

- 6) Buat *event* pada file *javascript* customDPQT, event jika *form* disubmit. Data akan dilanjutkan ke *controllerfunction* SaveGroup tadi.

```
$(document).on("submit", ".dpqt-group-formsave", function(e) {
    e.preventDefault()

    var formdata = new FormData(this)
    var result = InputRequired(this)

    if(result.cond == 1) {
        SwalInputRequired(result.col)
    } else {
        AjaxSaveForm("PenjualanQuickTruck/Group/SaveGroup", formdata)
    }
})
```

Gambar 3.117. Event Submit Form Menu Group

- 7) 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*.
- 8) Buka file *V_Group* tambahkan baris *html* untuk membuat sebuah tabel.

```
<table class="table table-bordered table-hover table-striped" id="dpqt-group-tablegroup">
  <thead class="bg-primary">
    <th class="text-center" style="width:20px">No</th>
    <th class="text-center">Group Name</th>
    <th class="text-center">Provinsi</th>
    <th class="text-center">Cabang</th>
    <th class="text-center" style="width:80px">Action</th>
  </thead>
</table>
```

Gambar 3.118. Kode *Html* Tabel Menu Group

- 9) Masih di file *V_Group* tambahkan baris script untuk meload data secara *serverSide* menggunakan *datatable*.

```
<script>
$(document).ready(() => {
  $("#dpqt-group-tablegroup").DataTable({
    processing: true,
    serverSide: true,
    pageLength: 10,
    order: [],
    ajax: {
      url: baseurl + "PenjualanQuickTruck/Group/ServerSideGroup",
      type: 'POST',
    },
    language: {
      processing: "<div class='overlay custom-loader-background'" +
        "<i class='fa fa-spinner fa-pulse' style='color:red'" +
        "</div>"
    },
  })
})
</script>
```

Gambar 3.119. *AjaxServerSide* Tabel

- 10) Buka file C_Group buat *function* *ServerSideGroup*, *function* ini digunakan untuk mengirimkan data ke *View*.













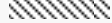


```
public function ServerSideGroup()
{
    $page['from'] = $_POST['start'] + 1;
    $page['to'] = $_POST['length'] + $_POST['start'];
    $filter = strtoupper($_POST['search']['value']);
    $result = $this->M_group->DataGroups($page, $filter);

    $data = [];
    foreach ($result as $value) {
        $row = [];
        $row[] = $value['NOMOR'];
        $row[] = $value['GROUP_NAME'];
        $row[] = $value['PROVINCE'];
        $row[] = $value['NAME'];
        $row[] = '<div class="btn-group" role="group">
            <button type="button"
                class="btn btn-success dpqt-group-button-showmodaledit access-update"
                title="Edit" data-id="' . $value['GROUP_ID'] . '"><i class="fa fa-edit"></i>
            </button>
            <button type="button"
                class="btn btn-danger dpqt-group-button-deletegroup access-delete"
                title="Delete" data-id="' . $value['GROUP_ID'] . '"
                data-name="' . $value['GROUP_NAME'] . '"><i class="fa fa-trash"></i>
            </button>
        </div>';
        $data[] = $row;
    }
    $output = array(
        "draw" => $_POST['draw'],
        "recordsTotal" => $this->M_group->CountAllGroup(),
        "recordsFiltered" => $this->M_group->CountFilteredGroup($filter),
        "data" => $data
    );
    echo json_encode($output);
}
```

Gambar 3.120. *FunctionServerSide* MenuGroup

- 11) Coba refresh / akses url ".../PenjualanQuickTruck/Group" maka pada tabel akan terdapat data yang pernah disimpan.

Show entries

No	Group Name	Provinsi	Cabang	Action
1	TRIAL T0014	Sumatera Barat	Pusat	 
2	TRIAL REBUILD DPQT	Jawa Tengah	Pusat	 
3	KLK	Kalimantan Timur	Banjarmasin	 
4	Trial Group	Jawa Tengah	Nganjuk	 
5	TRIAL 4	DKI Jakarta	Subang	 
6	TRIAL 1	Yogyakarta	Pusat	 
7		Bengkulu	Medan	 

Showing 1 to 7 of 7 entries

Previous
 1
 Next

Gambar 3.121. Tampilan Tabel Menu Group

- 12) Pada tabel diatas terdapat dua button pada kolom action, yaitu button edit/hijau dan button delete/merah.

- 13) Buka file V_Group, tambahkan baris *html* untuk membuat tampilan pop up.

```
<div class="modal fade" id="dpqt-group-modal-editgroup">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <form class="dpqt-group-form-edit-group">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-label="Close" style="margin:unset">
            <i class="fa fa-close"></i>
          </button>
          <h4 class="modal-title" id="modalLabelSmall">
            <b><i class="fa fa-edit text-primary mr-3"></i>Edit Group</b>
          </h4>
        </div>
        <div class="modal-body">
        </div>
        <div class="modal-footer" style="display:none">
          <button type="submit" class="btn btn-primary">Save changes</button>
        </div>
      </form>
    </div>
  </div>
</div>
```

Gambar 3.122. *Html* Modal Edit Group

- 14) Buat event pada file *javascript* customDPQT.js, event jika button edit terclick maka akan memunculkan pop up yang berisikan form edit.

```
$(document).on("click", ".dpqt-group-button-showmodaledit", function() {
  $("#dpqt-group-modal-editgroup").modal("show")

  ModalBody("PenjualanQuickTruck/Group/ModalEditGroup",
    $(this).data("id"), "#dpqt-group-modal-editgroup")
})
```

Gambar 3.123. *Event* Klik Tombol Edit Group

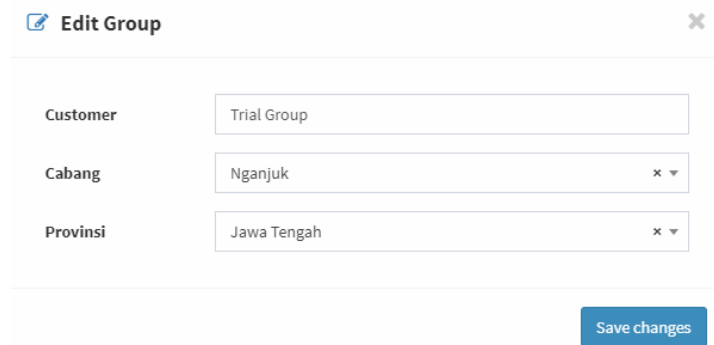
- 15) Buka file C_Group buat *function* ModalEditGroup, *function* ini digunakan untuk mengisi data data pop up.

```
public function ModalEditGroup()
{
  $group_id = $_GET['id'];
  $data = $this->M_group->GetGroupWithID($group_id);

  $modal_body =
  <div class="form-horizontal p-4">
    <div class="form-group">
      <label class="col-sm-3 control-label" style="text-align: left">Customer</label>
      <div class="col-sm-9">
        <input type="hidden" name="GROUP_ID" value=" $group_id . ' ">
        <input type="text" autocomplete="off" name="GROUP_NAME" value=" ' . $data['GROUP_NAME'] . ' " class="
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3 control-label" style="text-align: left">Cabang</label>
      <div class="col-sm-9">
        <select name="BRANCH" class="form-control dpqt-select-branch input-required" data-placeholder="Sele
          <option value=" ' . $data['BRANCH'] . ' " selected="selected"> ' . $data['CABANG'] . ' </option>
        </select>
      </div>
    </div>
    <div class="form-group mb-0">
      <label class="col-sm-3 control-label" style="text-align: left">Provinsi</label>
      <div class="col-sm-9">
        <select name="PROVINCE" data-placeholder="Select Provinsi" class="form-control dpqt-select-province
          <option value=" ' . $data['PROVINCE'] . ' "> ' . $data['PROVINCE'] . ' </option>
        </select>
      </div>
    </div>
  </div>
  ;
  echo $modal_body;
}
```

Gambar 3.124. *FunctionForm* Edit Group Modal

- 16) Coba klik salah satu button edit ditabel, maka akan muncul pop up edit group.



Gambar 3.125. Tampilan Modal Edit Group

- 17) Buat *function* lagi, *function* SaveEditGroup digunakan untuk menyimpan update data melalui pop up.

```
public function SaveEditGroup()
{
    $this->M_group->SaveEditGroup(
        $_POST['GROUP_ID'],
        $_POST['GROUP_NAME'],
        $_POST['BRANCH'],
        $_POST['PROVINCE'],
        $this->session->user
    );
    echo json_encode(1);
}
```

Gambar 3.126. *Function* Menyimpan Edit Group

- 18) Buat *event* pada file javascript customDPQT.js, *event* jika *form* popup tersubmit.

```
$(document).on("submit", ".dpqt-group-form-edit-group", function(e) {
    e.preventDefault()

    var formdata = new FormData(this)
    var result = InputRequired(this)

    if(result.cond == 1) {
        SwalInputRequired(result.col)
    } else {
        AjaxSaveForm("PenjualanQuickTruck/Group/SaveEditGroup", formdata)
    }
})
```

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

19) Fitur edit berhasil dibuat, selanjutnya buat fitur delete group.

Buka kembali C_Group, buat *function* DeleteGroup.

```
public function DeleteGroup()
{
    $this->M_group->DeleteGroup(
        $_GET['id'],
        $this->session->user
    );

    echo json_encode(1);
}
```

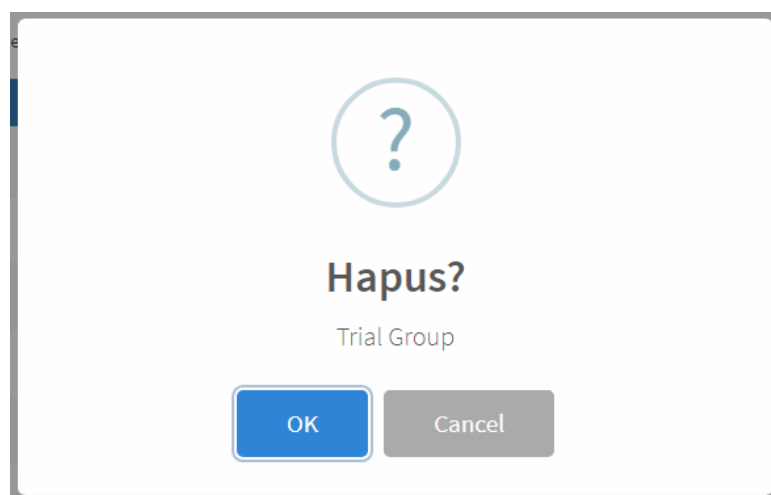
Gambar 3.128. *Function* Hapus Group

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

```
$(document).on("click", ".dpqt-group-button-deletigroup", function() {
    AjaxDelete($(this).data("name"),
        "PenjualanQuickTruck/Group/DeleteGroup" , $(this).data("id"))
})
```

Gambar 3.129. *Event* Klik Tombol Hapus Group

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



Gambar 3.130. Tampilan Pop-up Hapus Group

22) Menu Master Group telah selesai dibuat.


```

<div class="col-12 col-md-6 form-horizontal">
  <div class="form-group">
    <label class="col-sm-3 control-label">Provinsi</label>
    <div class="col-sm-9">
      <input type="text" autocomplete="off" class="form-control dpqt-customer-input-readonly" id="dpqt-customer-prov">
    </div>
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">No Telepon</label>
    <div class="col-sm-9">
      <input type="text" autocomplete="off" class="form-control" id="dpqt-customer-telephone" placeholder="No Telepon">
    </div>
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Nama No Telp</label>
    <div class="col-sm-9">
      <input type="text" autocomplete="off" class="form-control" placeholder="Nama No Telp (belum dipakai)" readonly="">
    </div>
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Jenis Pembeli</label>
    <div class="col-sm-9">
      <input type="text" autocomplete="off" class="form-control dpqt-customer-input-readonly" id="dpqt-customer-category">
    </div>
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Group</label>
    <div class="col-sm-9">
      <select class="form-control dpqt-select-group input-required" name="group_id" data-placeholder="Select Group">
        <option value=""></option>
      </select>
    </div>
  </div>
</div>
<div class="col-12 mt-4" style="display:flex">
  <button class="btn btn-primary m-auto"><b><i>Save Customer</i></b></button>
</div>
</form>

```

Gambar 3.134. *Htm/View* Menu Customer

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

Gambar 3.135. Tampilan Menu Customer

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

```

$(document).on("change", "#dpqt-customer-input-customer", function() {
  if($(this).val()) {
    $("#dpqt-customer-address").val($(this).select2("data")[0]['address'])
    $("#dpqt-customer-kecamatan").val($(this).select2("data")[0]['kecamatan'])
    $("#dpqt-customer-city").val($(this).select2("data")[0]['city'])
    $("#dpqt-customer-province").val($(this).select2("data")[0]['province'])
    $("#dpqt-customer-telephone").val($(this).select2("data")[0]['telephone'])
    $("#dpqt-customer-category").val($(this).select2("data")[0]['category'])
  } else {
    $("#dpqt-customer-input-readonly").val(null)
  }
})

```

Gambar 3.136. *Event* Kolom Customer Berubah

- 5) Kemudian buat *event* lagi, *event* jika *form* tersubmit.

```
$(document).on("submit", ".dpqt-customer-formsave", function(e) {
    e.preventDefault()

    var formdata = new FormData(this)
    var result = InputRequired(this)

    if(result.cond == 1) {
        SwalInputRequired(result.col)
    } else {
        cond = 0
        $(this).find('.dpqt-customer-input-readonly').each(function() {
            if($(this).val() == null || $(this).val() == '') {
                cond = 1
            }
        })

        if(cond == 1) {
            Swal.fire({
                type: 'error',
                title: 'Lengkapi Data!',
                html: 'Lengkapi data customer melalui aplikasi database customer'
            })
        } else {
            AjaxSaveForm("PenjualanQuickTruck/Customer/SaveCustomer", formdata)
        }
    }
})
```

Gambar 3.137. *Event Submit Form Menu Customer*

- 6) Lantas buka file *C_Customer*, buat *function* *SaveCustomer* yang akan digunakan menyimpan data ke *database*.

```
public function SaveCustomer()
{
    $cust_type = $_POST['customer_type'];
    $cust_account_id = ($_POST['cust_account_id']) ? $_POST['cust_account_id'] : '';
    $group_id = ($_POST['group_id']) ? $_POST['group_id'] : '';

    if ($cust_type == 'oracle') {
        $this->M_customer->SaveCustomerOracle($cust_account_id, $group_id, $this->session->user);
    } else if ($cust_type == 'potensial') {
        $this->M_customer->SaveCustomerPotential($cust_account_id, $group_id, $this->session->user);
    }

    echo json_encode(1);
}
```

Gambar 3.138. *Function Menyimpan Customer*

- 7) Fitur input customer sudah berhasil dibuat, sekarang buka file *V_Customer* dan tambahkan *html* untuk membuat tabel yang menampilkan data customer yang berhasil disimpan.

```
<table class="table table-bordered table-hover table-striped" id="dpqt-customer-table-customer">
    <thead class="bg-primary">
        <tr>
            <th class="text-center">No</th>
            <th class="text-center">Customer Name</th>
            <th class="text-center">Kecamatan</th>
            <th class="text-center">Contact Name</th>
            <th class="text-center">Contact Phone</th>
            <th class="text-center">Jenis Customer</th>
            <th class="text-center">Group</th>
            <th class="text-center" style="width:80px">Action</th>
        </tr>
    </thead>
</table>
```

Gambar 3.139. *Html Tabel Menu Customer*

- 8) Masih di V_Customer, tambahkan script *serverSide* untuk menampilkan data ke tabel.

```
<script>
$(document).ready(() => {
    $("#dpqt-customer-table-customer").DataTable({
        processing: true,
        serverSide: true,
        pageLength: 10,
        order: [],
        ajax: {
            url: baseurl + "PenjualanQuickTruck/Customer/ServerSideCustomer",
            type: 'POST',
        },
        language: {
            processing: "<div class='overlay custom-loader-background'>"+
                "<i class='fa fa-spinner fa-pulse' style='color:red'></i>"+
                "</div>"
        },
    })
})
</script>
```

Gambar 3.140. *AjaxServerSide* Tabel Customer

- 9) Kembali lagi ke file C_Customer, buat *function* *ServerSideCustomer* yang digunakan untuk mengirim data dari *database* ke table secara *serverSide* melalui *ajax* datatable.

```
public function ServerSideCustomer()
{
    $page['from'] = $_POST['start'] + 1;
    $page['to'] = $_POST['length'] + $_POST['start'];
    $filter = strtoupper($_POST['search']['value']);
    $result = $this->M_customer->DataCustomers($page, $filter);









    $data = [];
    foreach ($result as $value) {
        $row = [];
        $row[] = $value['NOMOR'];
        $row[] = $value['CUSTOMER'];
        $row[] = $value['KECAMATAN'];
        $row[] = '';
        $row[] = $value['CONTACT_PHONE'];
        $row[] = $value['JENIS_PEMBELI'];
        $row[] = $value['GROUP_NAME'];
        $row[] = '<div class="btn-group" role="group">
            <button type="button" class="btn btn-success dpqt-customer-button-showmodaledit access-update"
            title="Edit" data-id="' . $value['CUSTOMER_ID'] . '" data-type="' . $value['CUSTOMER_TYPE'] . '">
            <i class="fa fa-edit"></i></button>
            <button type="button" class="btn btn-danger dpqt-customer-button-deletecustomer access-delete"
            title="Delete" data-id="' . $value['CUSTOMER_ID'] . '" data-name="' . $value['CUSTOMER'] . '">
            <i class="fa fa-trash"></i></button>
        </div>';
        $data[] = $row;
    }
    $output = array(
        "draw" => $_POST['draw'],
        "recordsTotal" => $this->M_customer->CountAllCustomer(),
        "recordsFiltered" => $this->M_customer->CountFilteredCustomer($filter),
        "data" => $data
    );
    echo json_encode($output);
}
```

Gambar 3.141. *FunctionServerSide* Tabel Customer

- 10) Sekarang coba akses url menu Master Customer ".../PenjualanQuickTruck/Customer", jika tidak terdapat error

maka *serversidetabel* berhasil dibuat. Tampilan tabel seperti berikut.

Showing 1 to 4 of 4 entries

No	Customer Name	Kecamatan	Contact Name	Contact Phone	Jenis Customer	Group	Action
1		Kalasan			Perusahaan	TRIAL REBUILD DPQT	 
2		Kaway XVI			Perorangan	TRIAL 1	 
3		Pakem			Perorangan	ANGLO EASTERN	 
4		Krueng Barona Jaya			Perusahaan	TRIAL 1	 

Previous 1 Next

Gambar 3.142. Tampilan Tabel Customer

11) Di tabel terdapat dua tombol, button edit dan button delete.

Selanjutnya buat kedua fitur tersebut.

12) Buat fitur update terlebih dahulu, buat *event* pada *javascript* *customDPQT.js*. Event jika button edit yang berwarna hijau diclick maka akan memunculkan pop up.

```
$(document).on("click", ".dpqt-customer-button-showmodaledit", function() {
    $("#dpqt-customer-modal-editcustomer").modal("show")

    ModalBody("PenjualanQuickTruck/Customer/ModalEditCustomer",
    $(this).data("id")+'-'+$(this).data("type"), "#dpqt-customer-modal-editcustomer")
})
```

Gambar 3.143. *Event* Klik Tombol Edit Customer

13) Kemudian buka file *V_Customer*, tambahkan baris *html* untuk membuat tampilan pop up yang nantinya akan digunakan untuk *form* edit customer.

```
<div class="modal fade" id="dpqt-customer-modal-editcustomer">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <form id="dpqt-customer-form-edit-customer">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-label="Close" style="margin:unset">
            <i class="fa fa-close"></i>
          </button>
          <h4 class="modal-title" id="modalLabelSmall">
            <b><i class="fa fa-edit text-primary mr-3"></i>Edit Group Customer</b>
          </h4>
        </div>
        <div class="modal-body">
        </div>
        <div class="modal-footer" style="display:none">
          <button type="submit" class="btn btn-primary">Save changes</button>
        </div>
      </form>
    </div>
  </div>
</div>
```

Gambar 3.144. *Htm/View* Modal Edit Customer

- 14) Beralih ke file C_Customer, buat *function* ModalEditCustomer untuk menampilkan *form* beserta data customer yang akan dimunculkan ke pop up nantinya.

```

public function ModalEditCustomer()
{
    $param = explode('+', $_GET['id']);

    if ($param[1] == 'ORACLE') {
        $data = $this->M_customer->DetailCustomerOracle($param[0]);
    } else if ($param[1] == 'POTENSIAL') {
        $data = $this->M_customer->DetailCustomerPotential($param[0]);
    }

    $modal_body =
    '<div class="px-4 pb-2">
      <div class="p-4" style="background-color:#585764;color:white;border-radius:5px;display:flex">...
    </div>
    <div class="form-horizontal p-4">
      <div class="form-group">...
    </div>
    <div class="form-group">...
    </div>
    <div class="form-group">...
    </div>
    <div class="form-group">...
    </div>
    <div class="form-group">...
    </div>
    <div class="form-group">...
    </div>
    <div class="form-group">...
    </div>
    </div>
    ';
    echo $modal_body;
}

```

Gambar 3.145. *FunctionForm* Data Modal Edit Customer

- 15) Coba klik button edit, jika pop up sudah muncul beserta *formdata* berarti berhasil untuk memunculkan pop up.

Edit Group Customer

Jika ingin mengubah data customer dibawah, bisa langsung diubah melalui aplikasi Database Customer

Type: ORACLE

Customer: [Image Placeholder]

Kecamatan: Kalasan

Kota: Sleman (Kabupaten)

Provinsi: DI Yogyakarta

Kategori: Perusahaan

Group: TRIAL REBUILD DPQT

Save changes

Gambar 3.146. Tampilan Pop-up Edit Customer

- 16) Buat *event* pada *javascript* customDPQT.js jika *form* edit tersubmit.

```
$(document).on("submit", "#dpqt-customer-form-edit-customer", function(e) {
    e.preventDefault()

    var formdata = new FormData(this)
    var result = InputRequired(this)

    if(result.cond == 1){
        SwalInputRequired(result.col)
    } else {
        AjaxSaveForm("PenjualanQuickTruck/Customer/SaveEditCustomer", formdata)
    }
})
```

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

- 17) Beralih ke C_Customer, buat *function* SaveEditCustomer untuk menyimpan data edit dari *form* pop up.

```
public function SaveEditCustomer()
{
    $customer_id = $_POST['customer_id'];
    $old_group_id = $_POST['old_group_id'];
    $group_id = $_POST['group_id'];

    if ($group_id != $old_group_id) {
        $this->M_customer->SaveEditCustomer(
            $customer_id,
            $group_id,
            $this->session->user
        );
    }

    echo json_encode(1);
}
```

Gambar 3.148. *Function* Menyimpan Edit Customer

- 18) Fitur edit customer berhasil dibuat, sekarang buat fitur delete customer.
- 19) Buat *event* javascript customDPQT.js, *event* click button delete akan memunculkan pop up persetujuan delete.

```
$(document).on("click", ".dpqt-customer-button-deletecustomer", function() {
    AjaxDelete('Setup Customer : '+$(this).data('name'),
    "PenjualanQuickTruck/Customer/DeleteCustomer", $(this).data('id'))
})
```

Gambar 3.149. *Event* Klik Tombol Hapus Customer

20) Buka kembali file C_Customer, buat *function* DeleteCustomer.

Nantinya akan diteruskan ke *model* untuk diproses pada *database*.

```
public function DeleteCustomer()
{
    $this->M_customer->DeleteCustomer(
        $_GET['id'],
        $this->session->user
    );

    echo json_encode(1);
}
```

Gambar 3.150. *Function* Hapus Customer

21) Coba click button delete, jika pop up yang berisikan peringatan sudah muncul dan berjalan dengan lancar berarti fitur delete berhasil dibuat.

h. Membuat menu Input & Report Data Penjualan, *routes* yang dipakai:

```
$route['PenjualanQuickTruck/InputData'] = '\\\\C_Input';
$route['PenjualanQuickTruck/InputData/(:any)'] = '\\\\C_Input/$1';
```

Gambar 3.151. *Routes* Menu Input Data Penjualan

1) Buka file C_Input, buat *function* index untuk memanggil *view* V_Input.

```
public function index()
{
    $user_id = $this->session->userid;

    $data['Menu'] = 'Data Penjualan Quick Truck';
    $data['SubMenuOne'] = 'Input & Report Data Penjualan QT';
    $data['Title'] = 'Input Penjualan';
    $data['UserMenu'] = $this->M_user->getUserMenu($user_id, $this->session->responsibility_id);
    $data['UserSubMenuOne'] = $this->M_user->getMenuLv2($user_id, $this->session->responsibility_id);
    $data['UserSubMenuTwo'] = $this->M_user->getMenuLv3($user_id, $this->session->responsibility_id);
    $data['hak_akses'] = $this->UserAccess()['HAK_AKSES'];

    $this->load->view('V_Header', $data);
    $this->load->view('V_Sidemenu', $data);
    $this->load->view('PenjualanQuickTruck/Input/V_Input', $data);
    $this->load->view('V_Footer', $data);
}
```

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

- 2) Buka V_Input, tambahkan baris *html* untuk menampilkan *form* input data penjualan.

```
<form id="dpqt-input-formsave">
  <div class="row mx-0 pb-4">
    <div class="col-12 col-md-6 form-horizontal">
      <div class="form-group">...
    </div>
    <div class="form-group">
      <label class="col-sm-3 control-label">Customer</label>
      <div class="col-sm-9">
        <select class="form-control dpqt-select-customer input-required" name="customer_id"
          id="dpqt-input-select-customer" data-placeholder="Select Customer">
          <option value=""></option>
        </select>
        <input type="hidden">
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3 control-label">Nomor SO</label>
      <div class="col-sm-9">
        <select class="form-control dpqt-select-sonumber input-required" name="header_id_so"
          id="dpqt-input-select-sonumber" data-placeholder="Select SO Number" disabled>
          <option value=""></option>
        </select>
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3 control-label">Receipt Date</label>
      <div class="col-sm-9">
        <input type="text" autocomplete="off" name="receipt_date"
          class="form-control dpqt-input-datepicker input-required" placeholder="DD-MM-YYYY">
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3 control-label">Type</label>
      <div class="col-sm-9">
        <input type="text" autocomplete="off" class="form-control input-required"
          name="type" placeholder="Type">
      </div>
    </div>
  </div>
</form>
```

Gambar 3.152. *Html View Menu Input*

```
<div class="col-12 col-md-6 form-horizontal">
  <div class="form-group">
    <label class="col-sm-3 control-label">Safety Equip.</label>
    <div class="col-sm-9">...
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Cage Wheel</label>
    <div class="col-sm-9">...
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Side Trellis</label>
    <div class="col-sm-9">...
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Additional</label>
    <div class="col-sm-9">...
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Comments</label>
    <div class="col-sm-9">...
  </div>
</div>
<div class="col-12 mt-4" style="display:flex">
  <button class="btn btn-primary m-auto"><i class="fa fa-save mr-2"></i>Save</button>
</div>
</form>
```

Gambar 3.153. *Html View Menu Input*

- 3) Akses url “.../PenjualanQuickTruck/InputData”, maka akan tampil seperti ini.

Gambar 3.154. Tampilan *Form* Input

- 4) Buat event form submit pada *javascript* customDPQT.js.

```
$(document).on("submit", "#dpqt-input-formsave", function(e) {
    e.preventDefault()

    var formdata = new FormData(this)
    var result = InputRequired(this)

    if(result.cond == 1) {
        SwalInputRequired(result.col)
    } else {
        AjaxSaveForm("PenjualanQuickTruck/InputData/SaveOrder", formdata)
    }
})
```

Gambar 3.155. *Event* Submit *Form* Input

- 5) Lantas buka file *C_Input*, buat *function* *SaveOrder* untuk menangkap data dari *ajax* kemudian meneruskannya ke *model* untuk disimpan.

```
public function SaveOrder()
{
    $customer_id = $_POST['customer_id'];
    $header_id_so = $_POST['header_id_so'];
    $receipt_date = $_POST['receipt_date'];
    $type = $_POST['type'];
    $safety_equipment = (isset($_POST['safety_equipment'])) ? $_POST['safety_equipment'] : 'N';
    $cage_wheel = (isset($_POST['cage_wheel'])) ? $_POST['cage_wheel'] : 'N';
    $side_trellis = (isset($_POST['side_trellis'])) ? $_POST['side_trellis'] : 'N';
    $additional = $_POST['additional'];
    $comments = $_POST['comments'];

    $this->M_input->SaveOrder(
        $header_id_so,
        $customer_id,
        $receipt_date,
        $type,
        $safety_equipment,
        $cage_wheel,
        $side_trellis,
        $additional,
        $comments,
        $this->session->user
    );

    echo json_encode(1);
}
```

Gambar 3.156. *Function* Menyimpan *Form* Input

- 6) Fitur input order berhasil dibuat, sekarang buka file V_Input dan tambahkan baris *html* untuk membuat tabel yang akan menampilkan data order.

```
<table class="table table-bordered table-hover table-striped table-responsive" id="dpqt-input-table-order">
  <thead class="bg-primary">
    <th class="text-center bg-primary" style="white-space: nowrap;">No</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">No SO</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Customer</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Receipt Date</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Type</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Safety Equipment</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Cage Wheel</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Side Trellis</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Additional / Tambahan</th>
    <th class="text-center bg-primary" style="white-space: nowrap;">Comments</th>
  </thead>
</table>
```

Gambar 3.157. *Html* Tabel Input

- 7) Dan tambahkan script untuk menampilkan data tabel secara *server-side* menggunakan *ajax datatable*.

```
<script>
$(document).ready(() => {
  $('#dpqt-input-table-order').DataTable({
    dom: '<"#dpqt-input-table-order-dom">frtip',
    scrollX: true,
    scrollY: true,
    scrollCollapse: true,
    paging: true,
    fixedColumns: {
      leftColumns: 5,
    },
    processing: true,
    serverSide: true,
    pageLength: 10,
    order: [],
    ajax: {
      url: baseUrl + "PenjualanQuickTruck/InputData/ServerSideOrder",
      type: 'POST',
    },
    language: {
      processing: "<div class='overlay custom-loader-background'>"+
        "<i class='fa fa-spinner fa-pulse' style='color:red'></i>"+
        "</div>"
    },
  })
})
</script>
```

Gambar 3.158. *AjaxServerSide* Tabel Input

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

```
dom: '<"#dpqt-input-table-order-dom">frtip',
```

Gambar 3.159. DOM Datatable Tabel Input

- 9) buka file customDPQT.js tambahkan baris script untuk membuat tombol export report data penjualan.

```
$(document).ready() => {
    if($("#div#dpqt-input-table-order-dom").length > 0) {
        $.ajax({
            url: baseUrl + "PenjualanQuickTruck/InputData/GetCustomerCategory",
            dataType: "json",
            beforeSend: function() {
                $("#div#dpqt-input-table-order-dom").html(
                    '<div style="display:flex">' +
                    '<div class="mb-0 text-center"><i class="fa fa-cog fa-spin text-red"></i></div>' +
                    '</div>'
                );
            },
            success: function(response) {
                var option = '';
                $.each(response, function(index, value) {
                    option += '<option value="'+value.JENIS_PEMBELI+'">' + value.JENIS_PEMBELI + '</option>'
                });

                $("#div#dpqt-input-table-order-dom").html(
                    '<div class="access-export" style="display:flex">' +
                    '<div class="btn btn-danger mr-3" id="dpqt-input-button-reportdbqt"><i class="fa fa-file-excel-o mr-3"></i>' +
                    '<form class="input-group" id="dpqt-input-form-reportkonsumen" style="display:flex">' +
                    '<input autocomplete="off" type="text" class="form-control" name="tahun" style="border-color: #337AB7;' +
                    '<select class="form-control" name="type" style="border-color: #00A65A; border-left: none; border-right: none; border-top: none; border-bottom: none;">Pilih Jenis Customer</select>' +
                    '</form>' +
                    '</div>' +
                    '<div class="input-group btn btn-group p-0" role="group">' +
                    '<button type="submit" data-id="1" class="btn btn-success" title="Excel"><i class="fa fa-file-excel-o"></i>' +
                    '</div>' +
                    '</div>'
                );
            }
        });
    }
}
```

Gambar 3.160. Javascript Buat Tombol Export

- 10) Buat *function* GetCustomerCategory untuk option diatas.

```
public function GetCustomerCategory()
{
    $data = $this->M_input->SelectCustomerCategory();

    echo json_encode($data);
}
```

Gambar 3.161. Function List Kategori Customer

- 11) Buat juga *function* ServerSideOrder untuk menampilkan data pada table.

```
public function ServerSideOrder()
{
    $page['from'] = $_POST['start'] + 1;
    $page['to'] = $_POST['length'] + $_POST['start'];
    $filter = strtoupper($_POST['search']['value']);
    $result = $this->M_input->DataOrder($page, $filter);

    $data = [];
    foreach ($result as $value) {
        $row = [];
        $row[] = $value['NMOR'];
        $row[] = ($value['SO_NUMBER']) ? $value['SO_NUMBER'] : '-';
        $row[] = $value['CUSTOMER'];
        $row[] = ($value['RECEIPT_DATE']) ? $value['RECEIPT_DATE'] : '-';
        $row[] = $value['TYPE'];
        $row[] = ($value['SAFETY_EQUIP'] == 'Y') ? '<i class="fa fa-check text-green"></i>' : '<i class="fa fa-close text-red"></i>';
        $row[] = ($value['CAGE_WHEEL'] == 'Y') ? '<i class="fa fa-check text-green"></i>' : '<i class="fa fa-close text-red"></i>';
        $row[] = ($value['SIDE_TRELLIS'] == 'Y') ? '<i class="fa fa-check text-green"></i>' : '<i class="fa fa-close text-red"></i>';
        $row[] = $value['ADDITION'];
        $row[] = $value['COMMENTS'];
        $data[] = $row;
    }
    $output = array(
        "draw" => $_POST['draw'],
        "recordsTotal" => $this->M_input->CountAllOrder(),
        "recordsFiltered" => $this->M_input->CountFilteredOrder($filter),
        "data" => $data
    );
    echo json_encode($output);
}
```

Gambar 3.162. Function ServerSide Tabel Input

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

No	No SO	Customer	Receipt Date	Type	Safety Equipment	Cage Wheel	Side Trelis	Additional / Tan
1			19-JAN-22	QT-14	✓	✗	✗	Trial Addition
2			29-JAN-22	QT-14	✗	✗	✓	Additional
3			01-JAN-22	QT-14	✓	✗	✗	Addition
4	-		28-JAN-22	QT - 14 E	✗	✗	✗	Trial
5	-		-	QT 14	✗	✗	✗	cedsc
6	-		-	QT 14	✗	✗	✓	Triala

Gambar 3.163. Tampilan Tabel Input

- 13) Buat *event* click button report database qt pada *javascript* customDPQT.js.

```
$(document).on("click", "#dpqt-input-button-reportdbqt", function() {
    AjaxExportReport("PenjualanQuickTruck/InputData/ExportExcelDatabaseQT", '')
})
```

Gambar 3.164. Event Klik Tombol Export

- 14) Buka file C_Input, buat *function* ExportExcelDatabaseQT. *Function* ini berisi baris *code* untuk membuat file excel menggunakan phpxcel.

```
public function ExportExcelDatabaseQT()
{
    $objPHPExcel = new PHPExcel();

    $style_row = array(...);
    $style_center = array(...);
    $style_vercenter = array(...);

    $sheet = $objPHPExcel->getActiveSheet(0);

    $sheet->setCellValue('A1', 'DATABASE PENJUALAN QUICK TRUCK');
    $sheet->mergeCells('A1:Y1');
    $sheet->mergeCells('A2:Y2');

    $sheet->getRowDimension('1')->setRowHeight(60);
    $sheet->getRowDimension('2')->setRowHeight(22);
    $sheet->getRowDimension('3')->setRowHeight(22);
    $sheet->getDefaultrRowDimension()->setRowHeight(22);
    $sheet->getStyle('A1')->getFont()->setBold(true);
    $sheet->getStyle('A1')->getFont()->setSize(36);
    $sheet->getStyle('A1')->applyFromArray($style_center);

    $sheet->setCellValue('A3', "NO");
    $sheet->setCellValue('B3', "OPK SO");
    $sheet->setCellValue('C3', "SPB DO");
    $sheet->setCellValue('D3', "JUMLAH TRANSAKSI");
    $sheet->setCellValue('E3', "TOTAL AMOUNT");
    $sheet->setCellValue('F3', "CABANG");
    $sheet->setCellValue('G3', "CUSTOMER NAME");
    $sheet->setCellValue('H3', "ALAMAT");
    $sheet->setCellValue('I3', "KECAMATAN");
    $sheet->setCellValue('J3', "KOTA KECAMATAN");
    $sheet->setCellValue('K3', "PROVINSI");
}
```

Gambar 3.165. Function Export Excel Data Input

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

```
$route['PenjualanQuickTruck/DataInquiries'] = 'C_Inquiries';
$route['PenjualanQuickTruck/DataInquiries/(:any)'] = 'C_Inquiries/$1';
$route['PenjualanQuickTruck/DataInquiries/(:any)/(any)'] = 'C_Inquiries/$1/$2';
```

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 untuk 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 dahulu. Buka file *C_Inquiries* buat *function* index untuk memanggil view *V_Inquiries*.

```
public function index()
{
    $user_id = $this->session->userid;

    $data['Menu'] = 'Data Penjualan Quick Truck';
    $data['SubMenuOne'] = 'Input Data Inquiries QT';
    $data['tab'] = 'inquiries';
    $data['Title'] = 'Input Data Inquiries';
    $data['UserMenu'] = $this->M_user->getUserMenu($user_id, $this->session->responsibility_id);
    $data['UserSubMenuOne'] = $this->M_user->getMenuLv2($user_id, $this->session->responsibility_id);
    $data['UserSubMenuTwo'] = $this->M_user->getMenuLv3($user_id, $this->session->responsibility_id);
    $data['hak_akses'] = $this->UserAccess()['HAK_AKSES'];

    $this->load->view('V_Header', $data);
    $this->load->view('V_Sidemenu', $data);
    $this->load->view('PenjualanQuickTruck/Inquiries/V_Inquiries', $data);
    $this->load->view('V_Footer', $data);
}
```

Gambar 3.167. *Function* Memanggil View Menu Inquiries

- 5) Buka file view `V_Inquiries`, tambahkan baris *html* untuk membuat *form* input inquiries. *Form* ini digunakan untuk menginputkan data inquiries.

```
<form id="dpqt-inquiries-form-inputinquiries">
  <div class="row mx-0">
    <div class="col-12 col-md-6 form-horizontal">
      <div class="form-group">
        <label class="col-sm-3 control-label">Customer</label>
        <div class="col-sm-9">
          <select class="form-control input-required dpqt-select-customer"
            id="dpqt-inquiries-select-customers" name="customer_id" data-placeholder="Select Customer">
          </select>
          <input type="hidden" name="type" id="dpqt-inquiries-input-type" value="oracle">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-3 control-label">Cabang</label>
        <div class="col-sm-9">
          <input type="text" class="form-control input-required" id="dpqt-inquiries-input-branch"
            placeholder="Cabang" readonly>
          <input type="hidden" name="org_id" id="dpqt-inquiries-input-orgid">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-3 control-label">Status</label>
        <div class="col-sm-9">
          <select class="form-control input-required select2 dpqt-select2"
            id="dpqt-inquiries-input-status" name="status" data-placeholder="Select Status">
            <option></option>
            <option value="INQUIRIES">INQUIRIES</option>
            <option value="NO INQUIRIES">NO INQUIRIES</option>
          </select>
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-3 control-label">Produk</label>
        <div class="col-sm-9">
          <select class="form-control input-required dpqt-select-product dpqt-select2"
            id="dpqt-inquiries-input-product" name="inventory_item_id" data-placeholder="Select Products">
          </select>
        </div>
      </div>
    </div>
  </div>
</form>
```

Gambar 3.168. *Html View Menu Inquiries*

```
<div class="col-12 col-md-6 form-horizontal">
  <div class="form-group">
    <label class="col-sm-3 control-label">Tgl. Inquiries</label>
    <div class="col-sm-9">
      <input type="text" class="form-control input-required dpqt-input-datepicker" name="date"
        id="dpqt-inquiries-input-date" placeholder="Tanggal Inquiries" autocomplete="off">
    </div>
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Qty</label>
    <div class="col-sm-9">
      <input type="number" min=0 class="form-control input-required"
        id="dpqt-inquiries-input-qty" placeholder="Qty" name="qty">
    </div>
  </div>
  <div class="form-group">
    <label class="col-sm-3 control-label">Keterangan</label>
    <div class="col-sm-9">
      <textarea class="form-control input-required" style="height:83px"
        id="dpqt-inquiries-input-note" placeholder="Keterangan" name="notes"></textarea>
    </div>
  </div>
</div>
<div class="col-lg-12 mt-4" style="display:flex">
  <div class="m-auto">
    <input type="hidden" name="pre_order_id" id="dpqt-inquiries-input-preorderid">
    <button class="btn btn-primary mr-4">
      <i class="fa fa-save mr-3"></i>Save
    </button>
    <button type="button" class="btn btn-danger">
      <i class="fa fa-trash mr-3"></i>Clear
    </button>
  </div>
</div>
</form>
```

Gambar 3.169. *Html View Menu Inquiries*

- 6) Akses url “.../PenjualanQuickTruck/DataInquiries”, maka akan tampil seperti ini.

Gambar 3.170. Tampilan Menu *Form Inquiries*

- 7) Untuk proses save/menyimpan data sama dengan proses sebelumnya membuat *event* pada *javascript* dan membuat *function* pada controller, sehingga pada kali ini penulis tidak jelaskan.
- 8) 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*.

Tanggal Waktu Input Da		Excel	PDF	Search:				
No	Calon Pelanggan	Cabang	Deskripsi Permintaan	Tanggal Inquiries	Qty	Tag	Batal Inquiries	Action
1		KHS Pusat (OU)	QUICK TRUCK QT-14 E (2 LP, BAN STD) (KIT)	27-01-2022	1	ORACLE	Batal Inquiries	
2		KHS Pusat (OU)	QUICK TRUCK QT-14M FD-DRT (2 LP, SAFETY, DOUBLE REAR TIRE) (KIT), QT-14	19-01-2022	10	ORACLE	Batal Inquiries	

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 3.171. Tampilan Tabel Inquiries

- 9) 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.

- 10) Buat *event* click button Batal Inquiries pada *javascript* customDPQT.js.

```
$(document).on("click", ".dpqt-inquiries-button-cancelinquiries", function() {
    var id = $(this).data("id")
    var customer_name = $(this).parent().parent().find("td:nth-child(2)").html()
    var branch = $(this).parent().parent().find("td:nth-child(3)").html()
    var product = $(this).parent().parent().find("td:nth-child(4)").html()

    $("#dpqt-inquiries-modal-editinquiries form").removeAttr("id")
    $("#dpqt-inquiries-modal-editinquiries form").attr('id', 'dpqt-inquiries-form-cancel-inquiries')
    $("#dpqt-inquiries-modal-editinquiries .modal-title b").html('<i class="fa fa-ban text-warning mr-3"></i>Cancel Inquiries')

    $("#dpqt-inquiries-modal-editinquiries .modal-body").html( ...
    )

    $("#dpqt-inquiries-modal-editinquiries .modal-footer button").removeClass("btn-primary")
    $("#dpqt-inquiries-modal-editinquiries .modal-footer button").addClass("btn-warning")
    $("#dpqt-inquiries-modal-editinquiries .modal-footer button").html("Batalkan Inquiries")
    $("#dpqt-inquiries-modal-editinquiries .modal-footer").removeAttr("style")

    $("#dpqt-inquiries-modal-editinquiries").modal("show")
})
```

Gambar 3.172. *Event* Klik Tombol Cancel Inquiries

- 11) Jika button Batal Inquiries diclick maka akan muncul pop up seperti ini.

Gambar 3.173. Tampilan Pop-up Batal Inquiries

- 12) 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.

- 13) Untuk button report yang berada diatas tabel caranya sama dengan cara sebelumnya, tetapi disini menggunakan *form* submit. Jadi buat *eventform* submit untuk report.

```
$(document).on("click", "#dpqt-inquiries-form-report-inquiries button", function(e) {
    e.preventDefault()

    var id = $(this).data("id")
    var date = $("#dpqt-inquiries-form-report-inquiries").find('input').val()

    if(date == '') {
        SwalInputRequired("Tanggal Export")
    } else {
        if(id == 1) {
            AjaxExportReport("PenjualanQuickTruck/DataInquiries/ExportExcelInquiries/inquiries", date)
        } else if (id == 2) {
            AjaxExportReport("PenjualanQuickTruck/DataInquiries/ExportPDFInquiries/inquiries", date)
        }
    }
})
```

Gambar 3.174. Event Submit Export Data Inquiries

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

```
public function ExportPDFInquiries($cond)
{
    $date = $_GET['date'];
    $explodedate = explode('-', $date);

    $array = array();
    foreach ($branch as $value) {
        if ($cond == 'inquiries') {
            $datav = $this->M_inquiries->ReportDataInquiries(rtrim($explodedate[0]), ltrim($explodedate[1]), $value['ORGANIZATION']);
        } else if ($cond == 'custpotential') {
            $datav = $this->M_inquiries->ReportCustomerPotential(rtrim($explodedate[0]), ltrim($explodedate[1]), $value['ORGANIZATION']);
        }
        $row['branch'] = strtoupper($value['BRANCH']);
        $row['data'] = $datav;
        $array[] = $row;
    }

    $data['data'] = $array;

    $pdf = $this->pdf->load();
    $pdf = new mPDF('utf-8', array(297, 210), 0, 'calibri', 3, 3, 5, 0, 0, 0);
    $fill = $this->load->view('PenjualanQuickTruck/Inquiries/V_PdfInquiries', $data, true);
    $pdf->writeHTML($fill);

    ob_start();
    $pdf->output();
    $pdfData = ob_get_contents();
    ob_end_clean();

    $response = array(
        'filename' => $filename . date('d-m-Y His') . '.pdf',
        'file' => 'data:application/pdf;base64,' . base64_encode($pdfData)
    );

    die(json_encode($response));
}
```

Gambar 3.175. Function Export PDF Inquiries

- 15) Untuk *view* pdfnya, buka file *V_PdfInquiries*, dan tambahkan baris *html* untuk membuat tampilan sesuai format. Kalau excel









tampilannya digenerate dari code phpxcel, sedangkan untuk pdf menggunakan mpdf yang memanggil *view*.

- 16) 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.

Tanggal Waktu Input Da

ExcelPDF

Search:

No	Calon Pelanggan	Cabang	Deskripsi Permintaan	Tanggal Inquiries	Qty	Tag	Status	Action
1		KHS Pusat (OU)	QUICK TRUCK QT-14 E (2 LP, BAN LGPT) (KIT)	20-01-2022	7	ORACLE	NO INQUIRIES	 
2		KHS Pusat (OU)	QUICK TRUCK QT-14 E (2 LP, BAN LGPT) (KIT)	26-01-2022	10	POTENSIAL	NO INQUIRIES	 
3		KHS Pusat (OU)	QUICK TRUCK QT-14 E (2 LP, BAN CAMPUR (3 STD & 2 TR4)) (KIT)	10-01-2022	3	POTENSIAL	NO INQUIRIES	 
4		KHS Pusat (OU)	QUICK TRUCK QT-14 M (1 LP, BAN TR4) (KIT)	31-01-2022	1	ORACLE	NO INQUIRIES	 

Showing 1 to 4 of 4 entries

Previous1Next

Gambar 3.176. Tampilan Tabel Customer Potensial

- 17) Untuk fitur export, edit, deletenya juga sama persis dengan tab Input Inquiries.
- 18) Menu Input Data Inquiries berhasil dibuat.
- j. Menu Progress Rencana Penjualan juga mempunyai dua tab, yaitu tab Input Pre Order dan tab Data Series.
- k. Membuat tab Input Pre Order menu Progress Rencana Penjualan, *routes* yang dipakai:









```
$route['PenjualanQuickTruck/PreOrder'] = 'C_PreOrder';
$route['PenjualanQuickTruck/PreOrder/(.*)'] = 'C_PreOrder/$1';
```

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

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

Tanggal Waktu Input Da		Excel	PDF	Search:			
No	Calon Pelanggan	Cabang	Deskripsi Permintaan	Tanggal PO	Aplikasi	Supply Stock	Action
1		KHS Pusat (OU)	QUICK TRUCK QT-14M LARGE FLANGE (BAN AGR, SAFETY) (KIT)	26-01-2022	SAWIT	CABANG	 
2		KHS Pusat (OU)	QUICK TRUCK QT-14 E (2 LP, BAN CAMPUR (3 TR4 & 2 STD)) (KIT)	26-01-2022	SAWIT	PUSAT	 
3		KHS Pusat (OU)	QUICK TRUCK QT-14 E (2 LP, BAN LGPT) (KIT)	28-01-2022	SAWIT	PUSAT	 
4		KHS Pusat (OU)	QUICK TRUCK QT-14 E (2 LP, BAN STD) (KIT)	26-01-2022	SAWIT	PUSAT	 

Showing 1 to 4 of 4 entries

Previous 1 Next

Gambar 3.179. Tampilan Tabel Menu Pre Order

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

```
$route['PenjualanQuickTruck/DataSeries'] = 'C_PreOrder/DataSeries';
$route['PenjualanQuickTruck/DataSeries/(:any)'] = 'C_PreOrder/$1';
```

Gambar 3.180. *Routes* Tab Data Series

- 1) Buka file C_PreOrder, buat *function* *DataSeries* yang akan digunakan untuk memanggil view *V_DataSeries*.

```
public function DataSet()
{
    $user_id = $this->session->userid;

    $data['Menu'] = 'Data Penjualan Quick Truck';
    $data['SubMenuOne'] = 'Progress Rencana Penjualan QT';
    $data['tab'] = 'dataseries';
    $data['Title'] = 'Progress Rencana Penjualan';
    $data['UserMenu'] = $this->M_user->getUserMenu($user_id, $this->session->responsibility_id);
    $data['UserSubMenuOne'] = $this->M_user->getMenuLv2($user_id, $this->session->responsibility_id);
    $data['UserSubMenuTwo'] = $this->M_user->getMenuLv3($user_id, $this->session->responsibility_id);
    $data['hak_akses'] = $this->UserAccess("PenjualanQuickTruck/PreOrder")['HAK_AKSES'];

    $data['data'] = $this->M_preorder->GetDataSeries();

    $this->load->view('V_Header', $data);
    $this->load->view('V_Sidemenu', $data);
    $this->load->view('PenjualanQuickTruck/PreOrder/V_DataSeries', $data);
    $this->load->view('V_Footer', $data);
}
```

Gambar 3.180. *Function* Memanggil View Data Series

- 2) Kemudian buka *V_DataSeries*, buat tampilan *form* input data series.

```
<form id="dpqt-preorder-form-inputseries">
    <div class="col-12 col-md-6 form-horizontal">
        <div class="form-group mb-0">
            <label class="col-sm-3 control-label">Nama Series</label>
            <div class="col-sm-9">
                <input type="text" class="form-control input-required"
                name="series_name" placeholder="Nama Series" autocomplete="off">
            </div>
        </div>
        <br />
        <div class="form-group mb-0">
            <div class="col-sm-12" style="display:flex">
                <div class="m-auto">
                    <button type="submit" class="btn btn-primary mr-4">
                        <i class="fa fa-save mr-3"></i>Save
                    </button>
                    <button type="button" onclick="ClearInput('#dpqt-form-dataseries')" class="btn btn-danger">
                        <i class="fa fa-trash mr-3"></i>Clear
                    </button>
                </div>
            </div>
        </div>
    </div>
</form>
```

Gambar 3.181. *Html Form* Data Series

- 3) Akses url ".../PenjualanQuickTruck/DataSeries", maka akan seperti ini.

Nama Series

Nama Series

💾

 Save

🗑️

 Clear

Gambar 3.182. Tampilan *Form* Data Series

- 4) Buat *even* submit *form* input data series pada *javascript* customDPQT.js.

```
$(document).on("submit", "#dpqt-preorder-form-inputseries", function(e) {
    e.preventDefault()

    var formdata = new FormData(this)
    var result = InputRequired(this)

    if (result.cond == 1) {
        SwalInputRequired(result.col)
    } else {
        AjaxSaveForm("PenjualanQuickTruck/PreOrder/SaveSeries", formdata)
    }
})
```

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

- 5) Buka C_PreOrder kembali, buat *function* SaveSeries yang digunakan untuk menyimpan data series ke *database* melalui *model*.

```
public function SaveSeries()
{
    $this->M_preorder->SaveSeries(
        strtoupper($_POST['series_name']),
        $this->session->user
    );

    echo json_encode(1);
}
```

Gambar 3.184. *Function* Menyimpan Data Series

- 6) Kembali ke *view* V_DataSeries, tambahkan *html* untuk membuat tabel yang akan menampilkan data series.

```
<table class="table table-bordered table-hover table-striped table-responsive" id="dpqt-preorder-table-data-series">
<thead class="bg-primary">
<tr>
<th class="text-center" style="vertical-align:middle">No</th>
<th class="text-center" style="vertical-align:middle">Name Series</th>
<th class="text-center" style="vertical-align:middle">Created Date</th>
<th class="text-center" style="width:80px;vertical-align:middle">Action</th>
</tr>
</thead>
<tbody>
<?php
foreach ($data as $key => $value) { ?>
<tr>
<td class="text-center" style="vertical-align: middle;">?> $key + 1 ?></td>
<td class="text-center" style="vertical-align: middle;">?> $value['SERIES_NAME'] ?></td>
<td class="text-center" style="vertical-align: middle;">?> $value['CREATION_DATE'] ?></td>
<td class="text-center" style="vertical-align: middle;">
<div class="btn-group" role="group">
<button type="button" class="btn btn-success dpqt-preorder-button-showmodaleditdataseries access-update"
title="Edit" data-id="?> $value['SERIES_ID'] ?>" data-name="?> $value['SERIES_NAME'] ?>"
<i class="fa fa-edit"></i>
</button>
<button type="button" class="btn btn-danger dpqt-preorder-button-deletedataseries access-delete"
title="Delete" data-id="?> $value['SERIES_ID'] ?>" data-name="?> $value['SERIES_NAME'] ?>"
<i class="fa fa-trash"></i>
</button>
</div>
</td>
</tr>
<?php }
</tbody>
</table>
```

Gambar 3.185. *Html* Tabel Data Series





- 7) Cara diatas berbeda dengan cara-cara sebelumnya dalam menampilkan data tabel. Karen data series dikirim langsung melalui *function* *DataSet* pada *C_PreOrder*.

```
$data['data'] = $this->M_preorder->GetDataSet();
```

Gambar 3.186. Data *Array* Data Series

- 8) Tabel yang berhasil dibuat.

Show 10 entries Search:

No	Nama Series	Created Date	Action
1	PO QT-14 SERIES	23 DECEMBER 2021	 
2	PO QT-16 SERIES	23 DECEMBER 2021	 

Showing 1 to 2 of 2 entries Previous 1 Next

Gambar 3.187. Tampilan Tabel Data Series

- 9) Untuk fitur update dan delete sama dengan cara sebelum-sebelumnya.

m. Aplikasi Data Penjualan Quick Truck berhasil dibuat.

7. 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 memberikan perintah untuk mengupload aplikasi ke server.

8. Kesimpulan

Aplikasi Data Penjualan Quick Truck merupakan aplikasi yang digunakan untuk memudahkan user dalam mengurus 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.