

TUTORIAL BOOK

SISTEM KEWIRAUSAHAAN MAHASISWA

UNIVERSITAS LOGISTIK & BISNIS INTERNASIONAL

**M. FACHRIZA FARHAN
MUHAMMAD FARHAN AKBAR M.
NOVIANA RIZA**

SISTEM INFORMASI KEWIRAUSAHAAN MAHASISWA

Universitas Logistik & Bisnis Internasional

**NOVIANA RIZA
M FACHRIZA FARHAN
MUHAMMAD FARHAN AKBAR MUHLIS**



PT. Penerbit Buku Pedia
2023

SISTEM INFORMASI KEWIRAUSAHAAN MAHASISWA

Universitas Logistik & Bisnis Internasional

Penulis:

M Fachriza Farhan
Muhammad Farhan Akbar Muhlis
Noviana Riza

ISBN:

Editor:

Penyunting:

Desain sampul dan Tata letak:

Font:

Calibri

Penerbit:

PT. Penerbit Buku Pedia

Redaksi:

Athena Residence Blok. E No. 1, Desa Ciwaruga,
Kec. Parongpong, Kab. Bandung Barat 40559
Tel. 628-775-2000-300
Email : penerbit@bukupedia.co.id

Distributor:

Informatics Research Center
Jl. Sariasih No. 54
Bandung 40151
Email : irc@ulbi.ac.id

Cetakan Pertama, 2023

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apa pun tanpa ijin tertulis dari penerbit

PRAKATA

PRAKATA

Seiring dengan kemajuan teknologi yang terus berkembang di era ini turut memberi dampak di segala bidang pekerjaan. Teknologi yang ada saat ini telah mengalami transformasi sebagai bagian dari kehidupan manusia yang memberi banyak manfaat di berbagai bidang, salah satunya dalam pengembangan dan peningkatan kualitas kewirausahaan.

Menurut Keputusan Presiden RI no. 99 tahun 1998 pengertian usaha kecil adalah kegiatan ekonomi rakyat yang berskala kecil dengan bidang usaha yang secara mayoritas merupakan kegiatan usaha kecil dan perlu dilindungi untuk mencegah dari persaingan usaha yang tidak sehat. Dalam perkembangannya kehadiran teknologi dalam kemajuan kewirausahaan sangat diperlukan sebagai sarana pembantu guna mengefisienkan kinerja yang ada bagi usahawan. Melihat tingginya minat kewirausahaan mahasiswa Universitas Logistik & Bisnis Internasional khususnya pada program studi DIV-Teknik Informatika kewirausahaan membuat penulis merasa perlu menghadirkan sebuah rancangan sistem yang dapat memfasilitasi kegiatan jual beli serta pencatatan dan penyimpanan data penjualan produk, mahasiswa kampus Universitas Logistik & Bisnis Internasional agar dapat mengelola usahanya dengan optimal.

Adanya sistem ini kiranya dapat membantu upaya pengembangan kewirausahaan yang dilakukan mahasiswa guna mempermudah pembeli melakukan pembelian dan mempermudah penjual melakukan pencatatan pesanan, karena pesanan secara otomatis akan tercatat di dalam sistem. Dengan begitu penjual dapat mengetahui jenis dan jumlah pemesanan yang dilakukan pembeli serta memberi kemudahan kepada pembeli dalam melakukan proses pembelian. Sistem ini juga dapat memberi akses penambahan menu ataupun sebaliknya sehingga kedepannya penjual tidak perlu khawatir apabila ingin mengurangi atau menambahkan menu pada *web*.

Tujuan dari penulisan buku ini untuk memberitahu cara membuat sistem informasi wirausaha agar mempermudah proses pengolahan data menjadi lebih baik. Buku ini berisi tentang perancangan sistem informasi wirausaha agar lebih efisien dan efektif dalam melakukan proses pengolahan data baik barang masuk maupun barang keluar. Setiap bab yang disajikan

dalam buku ini disertakan juga dalam kode program yang bisa diakses melalui link.

DAFTAR ISI

PRAKATA	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	iv
1 BAB 1.....	1
1.1 APA ITU WIRAUSAHA?.....	1
1.2 MENGAPA MAHASISWA PERLU MENGETAHUI WIRAUSAHA?	2
1.3 APA MANFAAT WIRAUSAHA DALAM KEHIDUPAN MAHASISWA?	3
2 BAB 2.....	5
2.1 BAHASA PEMOGRAMAN.....	5
2.1.1 Front-end	6
2.1.2 HTML.....	6
2.1.3 CSS	7
2.1.4 JavaScript	8
2.1.5 Bootstrap	8
2.1.6 Back-end	9
2.1.7 PHP.....	10
2.2 APLIKASI YANG DIGUNAKAN	11
2.2.1 PROSES INSTALASI VISUAL STUDIO CODE	11
2.3 XAMP	15
2.3.1 INSTALASI XAMPP	15
2.4 PENYIMPANAN YANG DIGUNAKAN	19
2.4.1 MYSQL.....	20
3 BAB 3.....	21
3.1 ALUR	21

3.1.1	Proses Daftar Wirausaha	21
3.1.2	Proses Bisnis Login.....	23
3.2	TUTORIAL PEMBUATAN APLIKASI.....	23
3.2.1	PERANCANGAN <i>DATABASE</i>	24
3.2.2	PEMBUATAN KODE LOGIN,REGISTER DAN LOGOUT	30
3.2.3	PEMBUATAN CRUD(<i>CREATE,READ,UPDATE,DELETE</i>).....	35
4	BAB 4.....	58
5	INDEKS	64

DAFTAR GAMBAR

Gambar 1 Tampilan Wirausaha	2
Gambar 2 Tampilan front-end	6
Gambar 3 Logo HTML	6
Gambar 4 CSS	7
Gambar 5 JavaScript	8
Gambar 6 Bootstrap	8
Gambar 7 Back-end	9
Gambar 8 PHP	10
Gambar 9 Logo Visual Studio Code	11
Gambar 10 Tampilan website	11
Gambar 11 Tampilan license Agreement	12
Gambar 12 Tampilan select destination Location	13
Gambar 13 Tampilan select start menu folder	13
Gambar 14 Tampilan select add tasks	14
Gambar 15 Tampilan install	14
Gambar 16 Tampilan awal visual studio code	15
Gambar 17 Logo XAMPP	15
Gambar 18 Tampilan Website	16
Gambar 19 Tampilan setup XAMPP	16
Gambar 20 Tampilan select components	17
Gambar 21 Tampilan installation folder	17
Gambar 22 Tampilan language	18
Gambar 23 Tampilan ready to install	18
Gambar 24 Tampilan XAMPP	19
Gambar 25 Logo MySQLs	20
Gambar 26 Flowmap register	21
Gambar 27 Flow map login	23
Gambar 28 Tampilan design database	24
Gambar 29 Tampilan awal XAMPP	24
Gambar 30 Tampilan start apache dan mysql	25

Gambar 31 Tampilan localhost.....	25
Gambar 32 Tampilan membuat nama database	26
Gambar 33 Tampilan SQL	26
Gambar 34 Tampilan struktur database.....	29
Gambar 35 Tampilan tabel kategori	29
Gambar 36 Tampilan tabel produk.....	30
Gambar 37 Tampilan tabel users.....	30
Gambar 38 Kode koneksi ke database	30
Gambar 39 Kode program untuk login	31
Gambar 40 Kode untuk mengecek register	32
Gambar 41 Kode program session.	33
Gambar 42 Kode pemanggilan file	33
Gambar 43 Kode program untuk login.	34
Gambar 44 Kode logout.....	35
Gambar 45 Kode fungsi tambah.....	36
Gambar 46 Kode button diklik.....	37
Gambar 47 Kode untuk create produk.....	38
Gambar 48 Atribut enctype	38
Gambar 49 Kode untuk fungsi upload	39
Gambar 50 Kode pengecekan tombol	40
Gambar 51 Sintaks SQL untuk program read	40
Gambar 52 Kode untuk menampilkan data kategori	41
Gambar 53 Sintaks SQL untuk mengambil data produk.....	42
Gambar 54 Kode untuk menampilkan data produk.....	43
Gambar 55 Kode untuk menggunakan get.....	44
Gambar 56 Kode query ambil data.....	44
Gambar 57 Kode menampilkan data untuk di ubah part 1	45
Gambar 58 Kode menampilkan data untuk di ubah part 2	46
Gambar 59 Kode menampilkan data untuk di ubah part 3	46
Gambar 60 Kode menampilkan data untuk di ubah part 4	47
Gambar 61 Kode untuk bagian button	48
Gambar 62 Kode untuk mengambil data dari user	49

Gambar 63 Kode untuk mengubah data yang ada di database	50
Gambar 64 Kode untuk mengecek tombol.....	51
Gambar 65 Kode query data kategori	51
Gambar 66 Kode untuk fungsi ubah kategori.....	52
Gambar 67 Kode untuk menampilkan data yang sudah di input	53
Gambar 68 Kode untuk kondisi tombol.....	53
Gambar 69 Kode untuk mengarahkan ke file	54
Gambar 70 Kode query untuk mengambil data	54
Gambar 71 Kode fungsi hapus.....	55
Gambar 72 Kode untuk kondisi hapus.....	55
Gambar 73 Kode untuk mengarahkan ke file delete.....	56
Gambar 74 Query untuk mengambil data.....	56
Gambar 75 Kode fungsi hapus produk	57
Gambar 76 Kode untuk kondisi tombol hapus produk	57

BAB 1

MEMAHAMI LEBIH DEKAT APA ITU WIRAUSAHA?

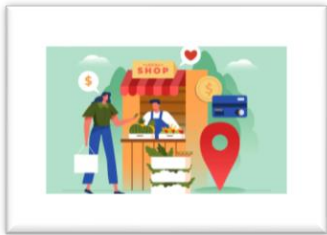
1.1 APA ITU WIRAUSAHA?

Pemahaman umum mengenai wirausaha yaitu menginisiasi usaha yang independen di bidang spesifik. Untuk menghindari kebingungan dalam definisi wirausaha, perlu dilakukan analisis secara menyeluruh guna mencapai pemahaman yang jelas tentang konsep wirausaha. Berikut beberapa definisi wirausaha menurut para pakar:

- a) Joseph Schumpeter dalam Buchori Alma (2011: 24) Wirausaha adalah orang yang mendobrak sistem ekonomi yang ada dengan memperkenalkan barang dan jasa yang baru dengan menciptakan bentuk organisasi baru atau mengolah bahan baku baru". Dari definisi di atas ditekankan bahwa seorang wirausaha adalah orang yang melihat peluang kemudian menciptakan sebuah organisasi baru dan memanfaatkan peluang tersebut.
- b) Menurut Kasmir (2011:19) Wirausahawan adalah orang yang berjiwa berani mengambil resiko untuk membuka usaha dalam berbagai kesempatan. Berjiwa berani mengambil resiko artinya bermental mandiri dan berani memulai usaha, tanpa diliputi rasa takut atau cemas sekalipun dalam kondisi tidak pasti.
- c) Menurut Suryana (2003:1) Wirausaha adalah kemampuan kreatif dan inovatif yang dijadikan dasar, kiat dan sumber daya untuk mencari peluang menuju sukses. Inti dari kewirausahaan adalah kemampuan untuk menciptakan sesuatu yang baru dan berbeda (create new and different) melalui berpikir kreatif dan bertindak inovatif untuk menciptakan peluang.

Dari penjelasan di atas, dapat disimpulkan bahwa seorang wirausahawan merupakan individu yang memulai usaha baru, mengambil peluang dan memanfaatkan waktu, modal, dan risiko, serta menerima imbalan sebagai balas jasa. Hal ini menunjukkan bahwa wirausaha tidak hanya bergantung pada modal semata.

1.2 MENGAPA MAHASISWA PERLU MENGETAHUI WIRAUSAHA?



Gambar 1 Tampilan Wirausaha

Memahami tentang wirausaha sangat penting dalam konteks ekonomi dan perkembangan bisnis terutama untuk mahasiswa. Berikut adalah beberapa alasan mengapa mahasiswa perlu memahami konsep wirausaha:

a) Peluang Karir

Pengetahuan tentang wirausaha akan membuka lebih banyak peluang karir. Mengetahui cara memulai dan mengelola bisnis sendiri dapat menawarkan banyak opsi karir selain menjadi karyawan. Mahasiswa dapat menjadi pengusaha sukses dan membangun bisnis mereka sendiri.

b) Kemandirian dan Inovasi

Kemandirian, inisiatif, dan inovasi adalah semua aspek yang diperlukan untuk menjadi wirausaha. Mengetahui wirausaha akan membantu siswa menjadi lebih kreatif, mandiri, dan berani menghadapi tantangan. Ini juga dapat meningkatkan kemampuan menyelesaikan masalah dan membuat keputusan.

c) Pemahaman Bisnis

Mahasiswa memperoleh pemahaman yang lebih baik tentang aspek-aspek bisnis karena memahami wirausaha. Mereka dapat mempelajari hal-hal seperti perencanaan bisnis, strategi pemasaran, manajemen keuangan, dan manajemen operasional, serta hal-hal lain yang berkaitan dengan menjalankan bisnis. Pengetahuan ini dapat diterapkan dalam berbagai situasi, seperti bekerja di perusahaan.

d) Pendorong Inovasi dan Pertumbuhan Ekonomi

Seringkali, inovasi dan pertumbuhan ekonomi didorong oleh wirausaha. Mahasiswa yang memiliki keterampilan wirausaha dapat membantu mengembangkan product, layanan, dan ide inovatif. Ini dapat

membantu pertumbuhan ekonomi suatu negara atau komunitas dengan menciptakan peluang baru, menciptakan lapangan kerja, dan banyak lagi.

e) Kemandirian Keuangan

Mahasiswa dapat belajar lebih banyak tentang manajemen keuangan pribadi dengan menjadi wirausaha. Mereka dapat mempelajari cara menghemat uang, investasi, dan pendapatan. Pengetahuan ini akan membantu mahasiswa mempelajari keterampilan keuangan yang bermanfaat sepanjang hidup mereka, terlepas dari keputusan mereka untuk berwirausaha atau tidak.

f) Pembelajaran Seumur Hidup

Bisnis membutuhkan pembelajaran dan adaptasi sepanjang hidup. Mengetahui bagaimana menjadi wirausaha akan membantu siswa mengembangkan mentalitas yang terus-menerus untuk belajar dan memperoleh kemampuan untuk menyesuaikan diri dengan perubahan pasar dan lingkungan bisnis yang cepat. Ini akan memberi mereka keuntungan di pasar kerja yang terus berkembang.

Oleh karena itu, pengetahuan tentang wirausaha memberikan mahasiswa pengetahuan dan perspektif yang bermanfaat untuk pengembangan pribadi dan profesional serta bisnis.

1.3 APA MANFAAT WIRAUSAHA DALAM KEHIDUPAN MAHASISWA?

Wirausaha memiliki banyak manfaat mahasiswa. Berikut adalah beberapa manfaat dari wirausaha untuk mahasiswa :

a) Kemandirian dan Pengembangan diri

Menjadi seorang wirausaha membantu siswa menjadi lebih mandiri dan mengambil tanggung jawab atas kehidupan mereka sendiri. Mereka akan belajar mengambil inisiatif, mengatasi tantangan, dan menguasai keterampilan problem-solving dan kepemimpinan.

b) Peluang Karir dan Penghasilan Potensial

Wirausaha menawarkan lebih banyak peluang karir dan lebih banyak uang. Mahasiswa yang wirausaha dapat membuat peluang kerja bagi diri mereka sendiri dan bahkan membantu orang lain. Mereka memiliki kemampuan untuk menghasilkan konsep bisnis yang berbeda dan inovatif, and mereka juga dapat memperoleh keuntungan moneter yang lebih besar daripada jika mereka hanya menjadi karyawan.

c) Kreativitas dan Inovasi

Mahasiswa menjadi lebih kreatif dan inovatif karena menjadi wirausaha. Mereka memiliki kemampuan untuk meneliti ide-ide baru, menghasilkan solusi unik, dan menghasilkan barang dan jasa yang memenuhi kebutuhan pasar yang belum terpenuhi. Hal ini memungkinkan siswa untuk menyampaikan diri dan mengembangkan kemampuan kreatif mereka.

d) Pembelajaran Praktis dan Pengalaman Kerja

Mahasiswa dapat memperoleh pembelajaran praktis di dunia bisnis dan pengalaman kerja langsung melalui wirausaha. Mereka dapat menghadapi masalah di dunia nyata, menerapkan pengetahuan yang mereka pelajari di kelas, dan belajar dari kegagalan dan kesuksesan mereka sendiri. Pengalaman ini dapat meningkatkan kemampuan mereka dan memberi mereka keunggulan di pasar kerja.

e) Jaringan dan Hubungan

Mahasiswa dapat membangun jaringan dan hubungan yang luas melalui wirausaha. Mereka memiliki kemampuan untuk berinteraksi dengan mentor, investor, pelanggan, sesama pengusaha, dan mitra bisnis potensial. Jaringan ini dapat memberikan akses ke sumber daya, peluang kolaborasi, dan dukungan yang berharga dalam pengembangan bisnis mereka.

f) Peningkatan Kemampuan Manajemen

Menjadi seorang wirausaha melibatkan pengembangan keterampilan manajemen yang komprehensif. Mahasiswa dapat belajar tentang perencanaan strategis, manajemen keuangan, manajemen operasional, pemasaran, dan keterampilan lainnya yang relevan untuk menjalankan bisnis. Keterampilan ini dapat diterapkan dalam berbagai konteks dan memberikan landasan yang kuat untuk pengembangan karir di masa depan.

Selain keuntungan yang disebutkan di atas, menjadi wirausaha juga dapat menyebabkan kepuasan pribadi dan memberi siswa kesempatan untuk mengambil kendali atas masa depan mereka. Mereka memiliki kemampuan untuk mewujudkan harapan mereka dan memberikan dampak positif pada masyarakat dan lingkungan sekitar dengan menjadi wirausaha.

BAB 2

PEMAHAMAN TENTANG SOFTWARE

2.1 BAHASA PEMOGRAMAN

Bahasa program adalah kumpulan petunjuk yang diberikan kepada komputer agar dapat melakukan tugas tertentu dan menyelesaikan masalah. Bahasa program membantu komputer mengolah data sesuai dengan langkah penyelesaian yang ditetapkan oleh programmer. Bahasa ini memungkinkan programmer untuk menentukan data mana yang akan diolah komputer, bagaimana data ini disimpan atau dikirim, dan jenis tindakan apa yang harus dilakukan dalam berbagai situasi. Pembuatan aplikasi wirausaha sendiri menggunakan dua jenis bahasa pemograman, yaitu *Front-end* dan *Back-end*. Kedua jenis bahasa pemograman ini memiliki peran penting masing-masing yang dapat membuat sistem dapat bekerja dengan baik. *Front-end* digunakan untuk memperindah tampilan agar *user* bisa lebih mudah berinteraksi dengan sistem. Sedangkan *Back-end* digunakan untuk mengatur sisi server dan sistem aplikasi.

Framework yang digunakan untuk *front-end* menggunakan *framework* css yang berfungsi untuk mempermudah *styling* pada bagian tampilan aplikasi. Sedangkan untuk bagian *back-end* menggunakan bahasa pemograman PHP murni atau orang-orang biasa menyebutnya dengan *PHP Native*.

2.1.1 Front-end

Front end mencakup segala sesuatu yang menghubungkan pengguna ke sistem back end. Biasanya berfungsi sebagai GUI untuk orang yang menggunakan sistem.



Gambar 2 Tampilan *front-end*

Front-end bekerja untuk membuat antarmuka pengguna yang menarik dan mudah digunakan. Ini termasuk desain yang menarik, navigasi yang mudah digunakan, dan interaksi yang responsif. Pengalaman pengguna yang menyenangkan saat berinteraksi dengan aplikasi atau situs *online* adalah tanggung jawab *front-end*. Bahasa pemrograman yang biasanya digunakan dalam pengembangan front end adalah html. (Goldbot, 2016:15). Mereka menggunakan bahasa pemrograman seperti HTML, CSS, dan JavaScript untuk membuat halaman web yang bersifat responsif dan menarik bagi pengguna. (carrerfoundry.com, n.d.).

2.1.2 HTML



Gambar 3 Logo HTML

Menurut Thabit (2011), Hypertext Markup Language 5 (HTML 5) adalah salah satu teknologi terbaru dalam standar World Wide Web. Standar ini memperkenalkan fitur baru seperti memutar video ataupun drag and drop,

sebelum ada HTML5, fitur untuk memutar video bergantung pada plugins pihak ketiga di penjelajah website seperti Adobe Flash atau Microsoft Silverlight. HTML5 membuat mekanisme yang lebih mudah dalam pencitraan, baik itu hanya audio, video, dua dimensi hingga tiga dimensi. Teknologi yang terbaru dari HTML5 adalah adanya fitur WebBased Graphic Library (WebGL) yang diadaptasi dari aplikasi desktop agar website dapat menjalankan sebuah Interaktif tiga Dimensi yang dimasukkan dalam JavaScript Object Notation (JSON). JSON adalah suatu format ringkas pertukaran data dalam komputer yang digunakan untuk merepresentasikan struktur data sederhana dan sebuah objek yang didasarkan JavaScript dan tidak bergantung pada suatu bahasa

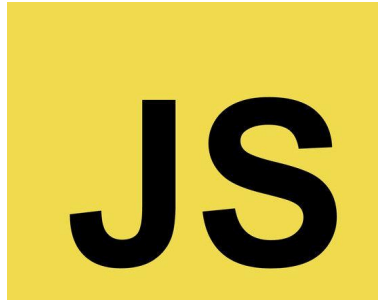
2.1.3 CSS



Gambar 4 CSS

CSS atau *Cascading Style Sheet* adalah bahasa yang digunakan untuk mempresentasikan halaman web dengan cara yang menarik dan atraktif. Representasi tersebut meliputi warna, tata letak, dan font. CSS memungkinkan web developer untuk membuat halaman web yang dapat menyesuaikan dengan berbagai ukuran layar. Biasanya, skrip CSS disimpan terpisah dari halaman HTML, namun juga bisa disisipkan dalam halaman HTML untuk mempermudah pengaturan halaman yang memiliki rancangan yang sama. (Duckett, 2011).

2.1.4 JavaScript



Gambar 5 JavaScript

JavaScript adalah bahasa pemrograman berbentuk kumpulan script yang berjalan pada suatu dokumen HTML (Lavarino & Yustanti, 2016, p. 74). Sejarah JavaScript dimulai pada tahun 1994, saat internet dan website berkembang pesat. Pada saat itu, bahasa pemrograman PERL biasanya digunakan untuk membuat *web*, dan pemrosesan hanya dapat dilakukan di sisi *web* server.

Salah satu kelemahan pemrosesan *web* server adalah bahwa arahan pengguna harus dikirim terlebih dahulu ke web server, baru kemudian ditampilkan di *web* browser. Hal ini dianggap tidak efektif karena kecepatan koneksi internet yang rendah. Programmer *web* membutuhkan bahasa pemrograman client-side yang dapat digunakan dalam browser *web* tanpa dikirim ke server.

2.1.5 Bootstrap



Gambar 6 Bootstrap

Bootstrap adalah paket aplikasi siap pakai untuk membuat front-end sebuah website. Bisa dikatakan, bootstrap adalah template desain web dengan fitur plus. Bootstrap diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai HTML dan CSS, anda pun siap menggunakan bootstrap(Rozi A. Zaenal & Comunity Smit Dev, 2015).

2.1.6 Back-end

Back end, juga dikenal sebagai sisi server, adalah tempat proses aplikasi atau system berjalan. Di sini, data dapat ditambahkan, diubah, atau dihapus. Database dan server adalah contoh komponen back end yang menangani hal-hal yang biasanya tidak dilihat atau dihubungi langsung oleh pengguna.



Gambar 7 *Back-end*

Sudah sedikit dibahas tentang pekerjaan back end developer saat menjelaskan pengertiannya. Apakah itu semua? Itu pasti tidak. Di bawah ini terdapat penjelasan lebih lanjut tentang pekerjaan yang harus dilakukan oleh seorang back end developer.

2.1.6.1 Melakukan Perancangan Database

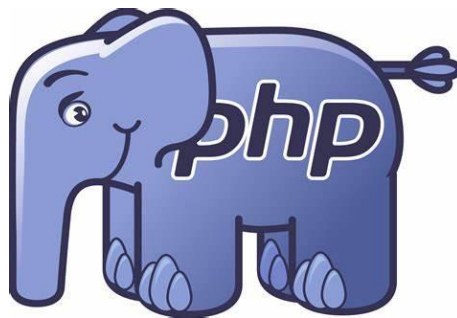
Data yang berkaitan dengan aplikasi atau website disimpan di database. Data yang dimaksud termasuk data pengguna, data produk, dan sebagainya. Oleh karena itu, sebelum membuat aplikasi atau situs online, database harus disiapkan terlebih dahulu. Tugas pertama bagi pengembang back end untuk

membuat database yang efisien sehingga tidak memberatkan server. Keamanan tidak boleh dikesampingkan juga. Data harus dibuat seaman mungkin.

2.1.6.2 Mengtasi Masalah yang Muncul di Back-end

Masalah mungkin muncul di sisi server atau back end software. Jika ini terjadi, tanggung jawab back end developer adalah melakukan troubleshooting. Untuk menghindari kesalahan yang sama, masalah harus diidentifikasi dan kemudian diperbaiki.

2.1.7 PHP



Gambar 8 PHP

Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat website yang bersifat server-side scripting. PHP bersifat dinamis. PHP dapat dijalankan pada berbagai macam sistem operasi seperti Windows, Linux, dan Mac Os. Selain Apache, PHP juga mendukung beberapa web server lain, seperti Microsoft ISS, Caudium, dan PWS. PHP dapat memanfaatkan database untuk menghasilkan halaman web yang dinamis. Sistem manajemen database yang sering digunakan bersama PHP adalah MYSQL. Namun, PHP juga mendukung sistem manajemen Database Oracle, Microsoft Access, Interbase, d-Base, dan PostgreSQL. (Andi, Wahana Komputer, 2014:h,73).

2.2 APLIKASI YANG DIGUNAKAN



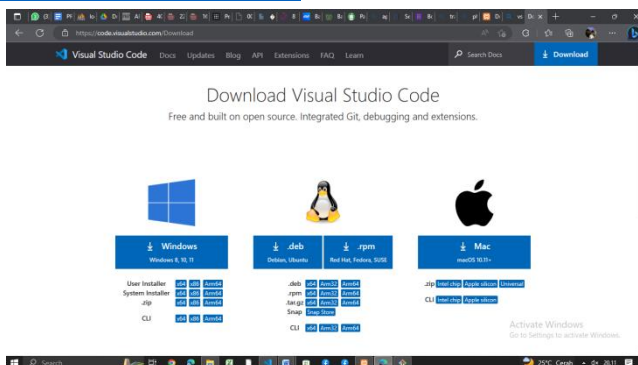
Gambar 9 Logo *Visual Studio Code*

Sebelum menggunakan bahasa pemrograman, sangat penting bagi seorang pengembang untuk memahami alat atau alat yang digunakan selama proses pengembangan. Salah satu alat yang digunakan saat membuat aplikasi helpdesk adalah Visual Studio Code sebagai *editor* teks dan XAMPP untuk menjalankan aplikasi di lingkungan lokal. Memahami dan memahami alat yang digunakan akan membantu proses pengembangan.

2.2.1 PROSES INSTALASI VISUAL STUDIO CODE

1. Kunjungi *website* resmi visual studio code

<https://code.visualstudio.com>.

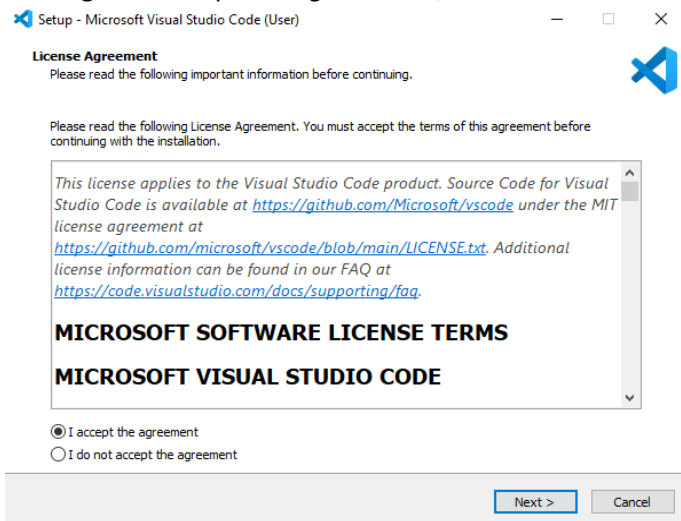


Gambar 10 Tampilan *website*

2. Di halaman utama, Anda akan melihat tombol "Unduh untuk Windows" jika Anda menggunakan Windows atau tombol "Unduh untuk macOS" jika

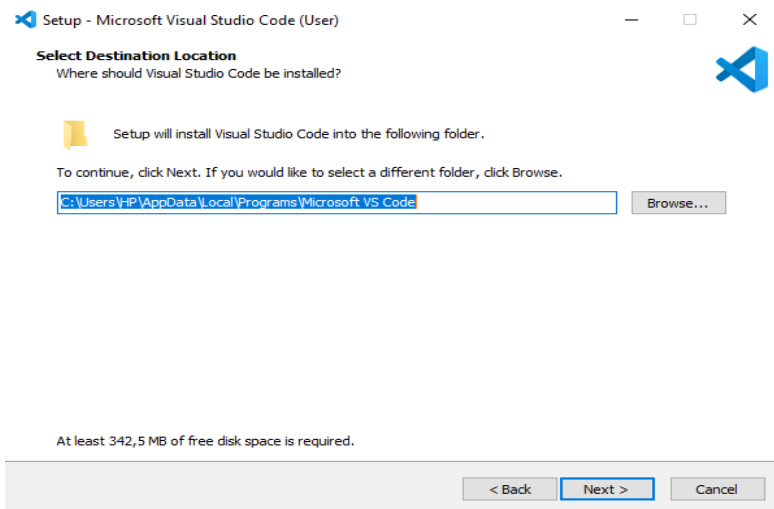
Anda menggunakan macOS. Klik tombol yang sesuai dengan sistem operasi Anda.

3. Setelah unduhan selesai, temukan *file* yang diunduh di sistem Anda.
4. Buka file yang sudah di download, maka akan menjadi seperti gambar di bawah, pilih bagian "*I accept the agreement*", lalu klik **Next**.



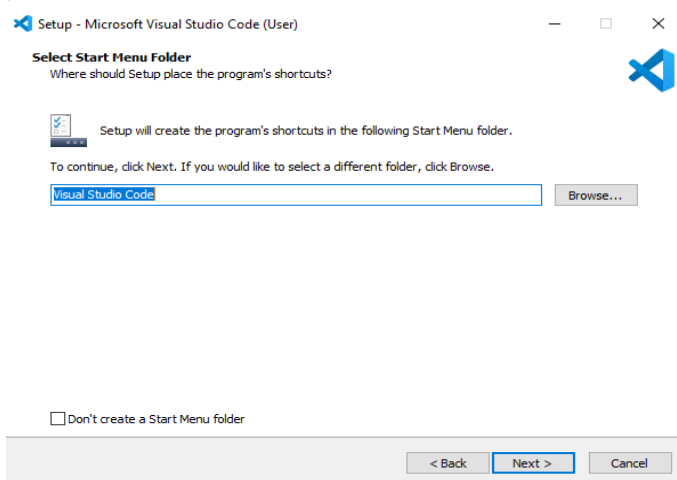
Gambar 11 Tampilan *license Agreement*

5. Setelah itu, jika muncul tampilan *Select Destination Location* seperti gambar di bawah, klik **Next**.



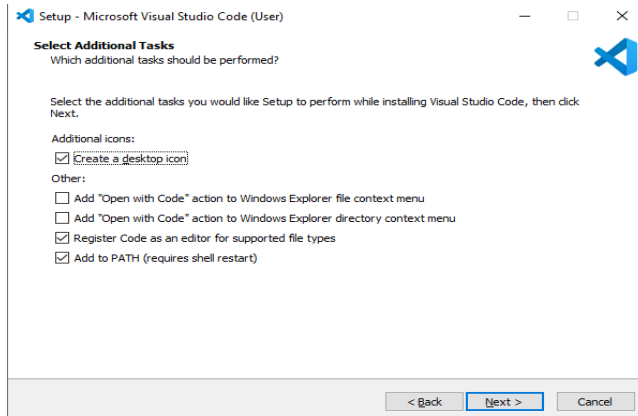
Gambar 12 Tampilan *select destination Location*

6. Kemudian, jika muncul tampilan *Select Start Menu Folder* seperti gambar di bawah, klik **Next**



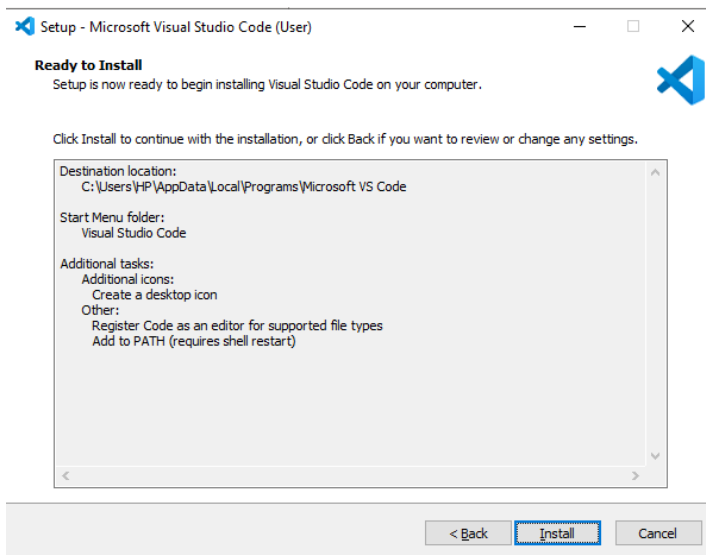
Gambar 13 Tampilan *select start menu folder*

7. Selanjutnya, jika muncul tampilan *Select Additional Tasks* seperti gambar di bawah, pilih bagian "*Create a desktop icon*", lalu klik **Next**.



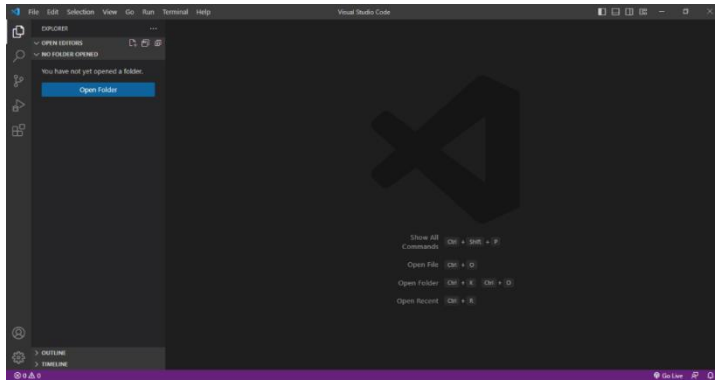
Gambar 14 Tampilan *select add tasks*

8. Setelah itu, apabila muncul tampilan seperti gambar di bawah, klik ***Install*** dan tunggu sampai proses instalasi selesai.



Gambar 15 Tampilan *install*

9. Terakhir, jika proses instalasi sudah selesai, buka aplikasi Visual Studio Code dan tampilan awal aplikasinya seperti gambar di bawah.



Gambar 16 Tampilan awal visual studio code

2.3 XAMP



Gambar 17 Logo XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Alan Nur Aditya, 2011:16).

2.3.1 INSTALASI XAMPP

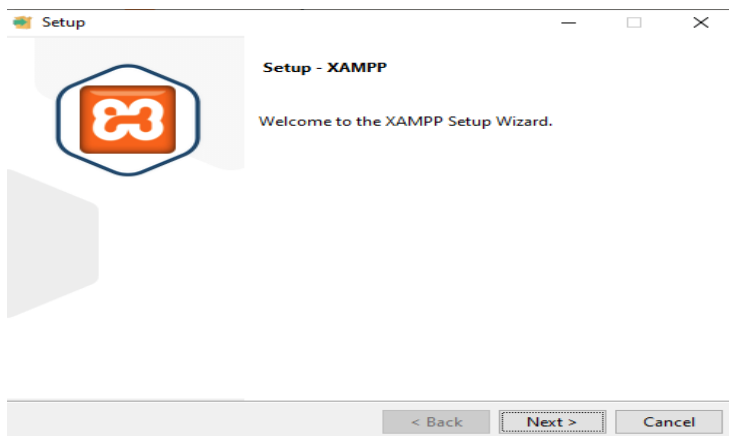
1. Kunjungi situs web resmi XAMPP di <https://www.apachefriends.org/index.html>
2. Pada halaman utama Anda akan melihat opsi untuk mengunduh XAMPP untuk berbagai platform seperti Windows, macOS, Linux, dan lainnya. Pilih

versi yang sesuai dengan sistem operasi operasi yang anda gunakan seperti gambar di bawah ini.



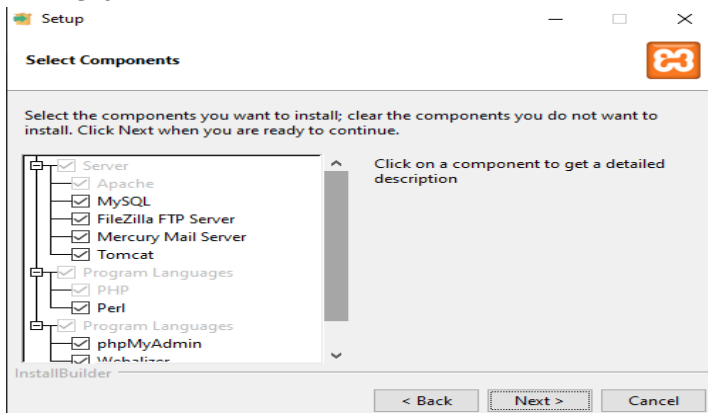
Gambar 18 Tampilan Website

3. Klik tombol unduhan untuk versi yang dipilih. Unduhan akan dimulai secara otomatis.
4. Setelah unduhan selesai, temukan *file installer* XAMPP di sistem Anda.
5. Jalankan *file installer* XAMPP dengan mengklik dua kali.
6. Jika sudah masuk tampilan *setup* XAMPP seperti gambar di bawah ini, klik **Next**.



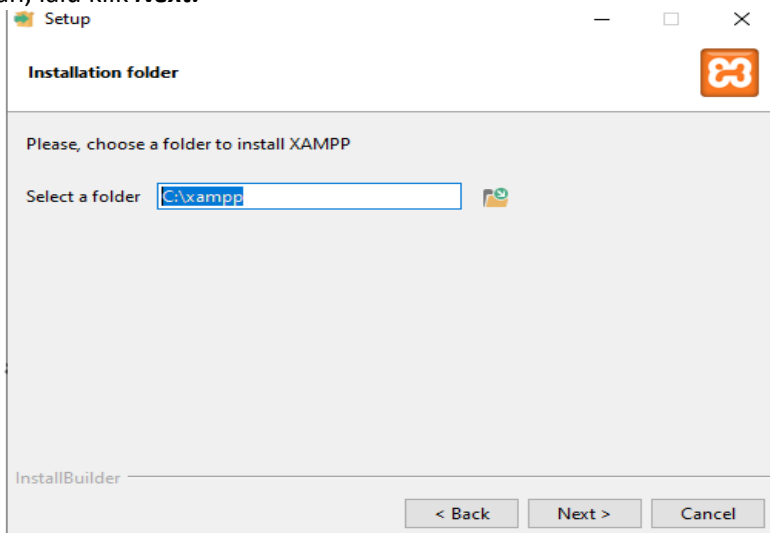
Gambar 19 Tampilan setup XAMPP

7. Kemudian, jika muncul tampilan *Select Components* seperti gambar di bawah, klik **Next**.



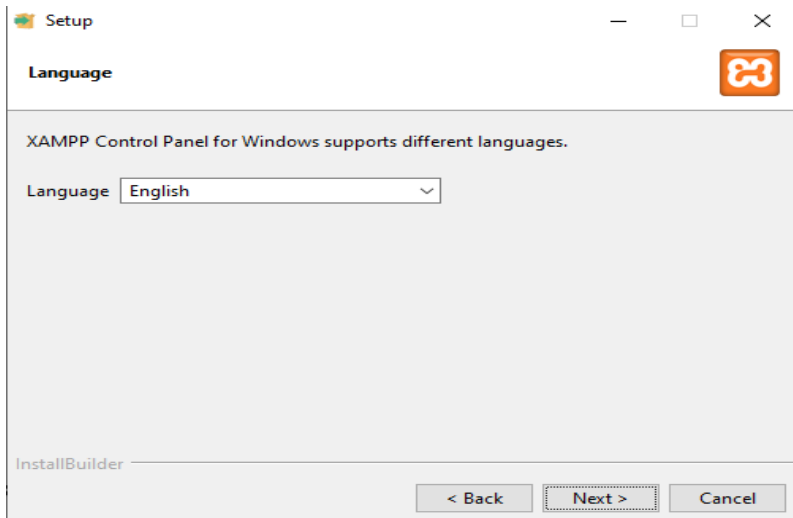
Gambar 20 Tampilan *select components*

8. Setelah itu, akan muncul tampilan *Installation Folder* seperti gambar di bawah, lalu klik **Next**.



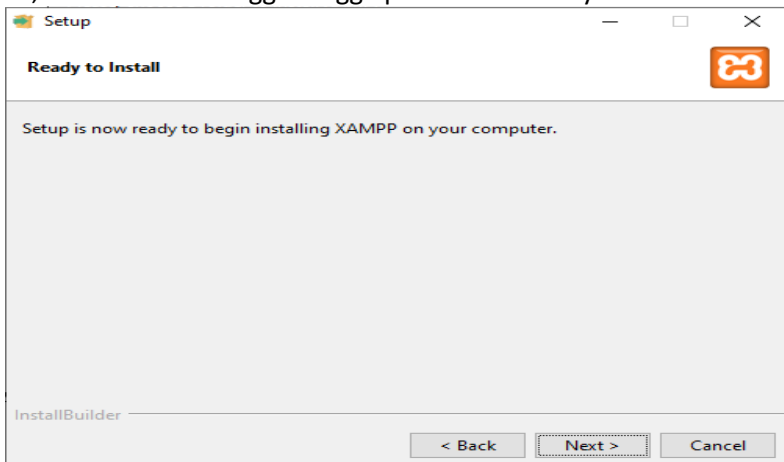
Gambar 21 Tampilan *installation folder*

9. Selanjutnya, jika muncul tampilan *Language* seperti gambar di bawah, klik **Next**.



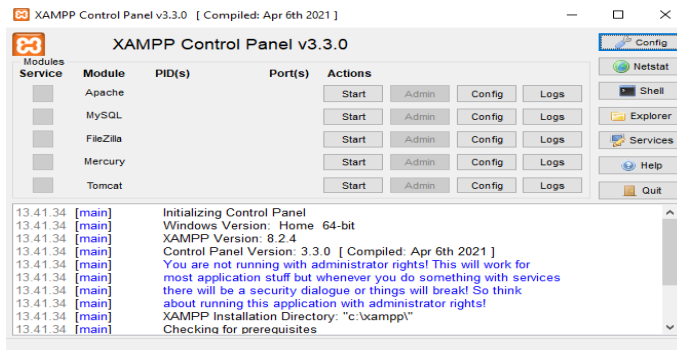
Gambar 22 Tampilan *language*

10. Kemudian, akan muncul tampilan *Ready to Install* seperti gambar di bawah, klik **Next** dan tunggu hingga proses instalasinya selesai.



Gambar 23 Tampilan *ready to install*

11. Setelah proses instalasi selesai, tampilan aplikasi XAMPP seperti gambar di bawah.



Gambar 24 Tampilan XAMPP

2.4 PENYIMPANAN YANG DIGUNAKAN

Penyimpanan atau yang biasa kita sebut dengan *database* menurut ahli yang ada pada buku Simarmata & Paryudi (2006:1), sebagai berikut:

- Menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data.
- Menurut silberschatz, dkk (2002) mendefenisikan basisdata sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan.
- Menurut Ramakrishnan dan Gehrke (2003) menyatakan basisdata sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.
- Menurut McLeod, dkk (2001), adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi.

Dari pendapat di atas pengertian *database* adalah *system* atau mekanisme yang digunakan untuk menyimpan dan mengorganisir kumpulan data yang sesuai dengan kebutuhan organisasi atau perusahaan. Semua sumber daya berbasis komputer yang dimiliki oleh organisasi tergabung dalam basis data, yang digunakan untuk menunjukkan, mengawasi, dan memanipulasi tindakan yang terkait.

Aplikasi wirausaha membutuhkan database atau penyimpanan sebagai cara untuk menyimpan keluhan atau masalah dan menampung data yang diperlukan. Oleh karena itu, kita akan menggunakan aplikasi database MYSQL. Mari kita mengenal MYSQL.

2.4.1 MYSQL



Gambar 25 Logo MySQLs

Menurut Rulianto Kurniawan (2010 :16) MySQL merupakan suatu jenis database server yang sangat terkenal. MySQL termasuk jenis RDBMS (*Relational Database Manajement System*). MySQL mendukung bahasa pemrograman PHP, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. MySQL merupakan RDBMS (*Relational Database Management System*) server.

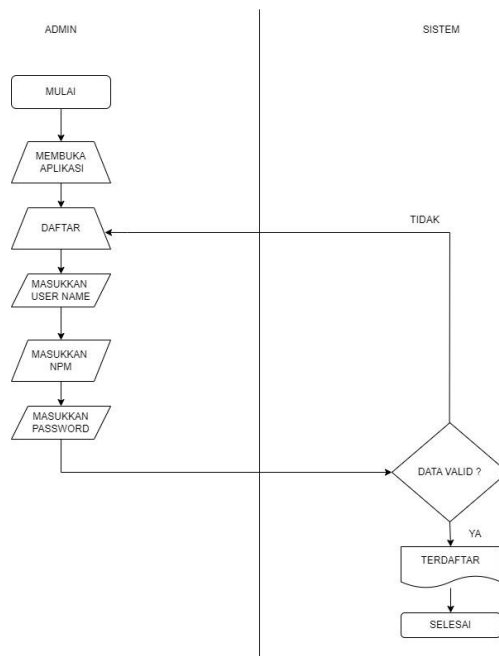
RDBMS adalah sebuah aplikasi yang memungkinkan pengguna database membuat, mengelola, dan menggunakan data dalam model hubungan. Oleh karena itu, semua tabel dalam database memiliki hubungan satu sama lain.

BAB 3

ALUR DAN TUTORIAL PEMBUATAN SISTEM

3.1 ALUR

3.1.1 Proses Daftar Wirausaha

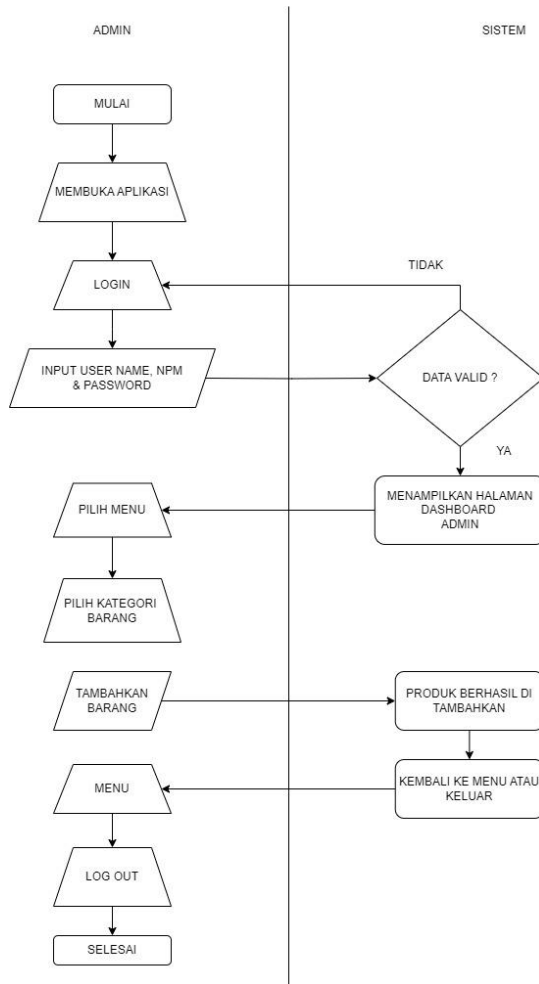


Gambar 26 Flowmap register

Keterangan:

- 1 Sebelum mempunyai akun , *user* harus melakukan daftar terlebih dahulu
- 2 Lalu masukan *username* atau nama lengkap , password , npm
- 3 Ketika datanya tidak sama maka akan di kembalikan kembali ke halaman daftar
- 4 Ketika berhasil, maka akun terbuat

3.1.2 Proses Bisnis Login



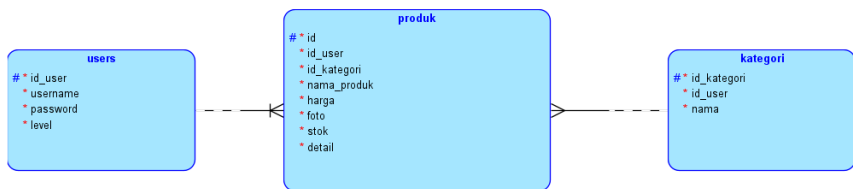
Gambar 27 Flow map login

3.2 TUTORIAL PEMBUATAN APLIKASI

Untuk melakukan pembuatan aplikasinya kita terlebih dahulu membuat rancangan database kemudian dilanjutkan dengan pembuatan kode program untuk menjalankan aplikasi.

3.2.1 PERANCANGAN DATABASE

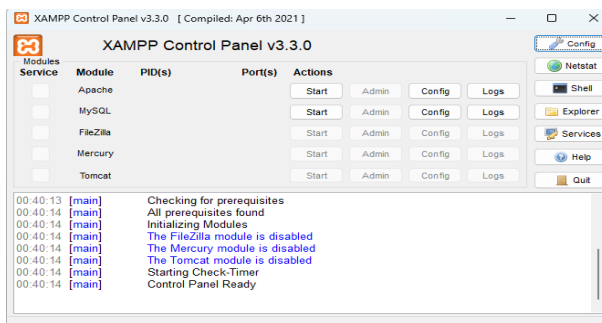
Saat kita membuat aplikasi, lebih baik membuat rancangan *database* untuk mengetahui bagaimana *database* kita berhubungan satu sama lain. Apakah ada alasan mengapa merancang *database* sangat penting? Rancangan *database* digunakan untuk mengetahui bagaimana alur kerja aplikasi kita akan berjalan akan ditentukan oleh desain *database*. Aplikasi yang baik juga berasal dari *database* yang baik.



Gambar 28 Tampilan *design database*

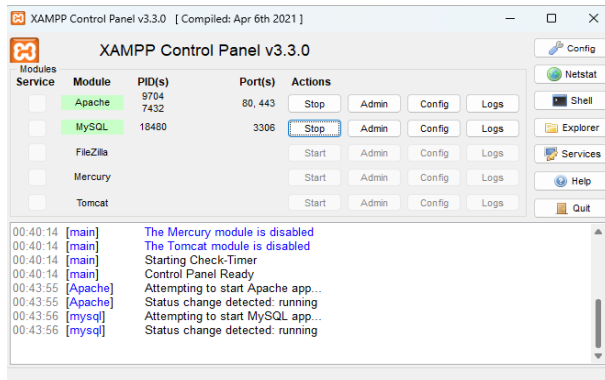
Untuk membuat *design database* seperti pada gambar 3.1 di atas kita memerlukan software yang bernama XAMPP. Dengan menggunakan XAMPP kita dapat membuat *database* melalui phpmyadmin. Adapun langkah-langkah yang harus dilakukan agar dapat membuat *database*.

1. Buka tampilan XAMPP seperti gambar dibawah



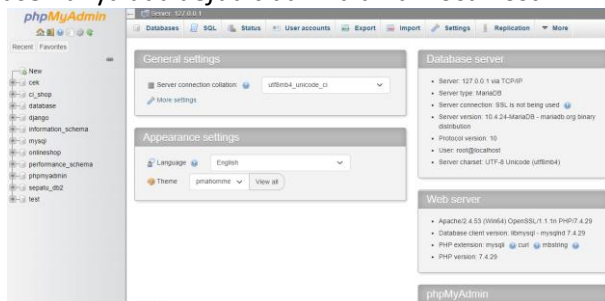
Gambar 29 Tampilan awal XAMPP

2. Kemudian nyalakan bagian Apache dan MYSQL



Gambar 30 Tampilan start apache dan mysql

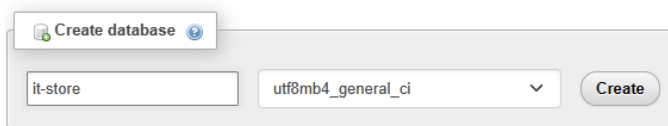
3. Lalu tekan tombol admin pada bar mysql nanti akan muncul halaman *localhost* yang nanti akan menampilkan beberapa *database* yang anda miliki, jika anda belum mempunyai *database* maka belum ada *database* hanya ada *default* dari halaman *localhost*



Gambar 31 Tampilan localhost

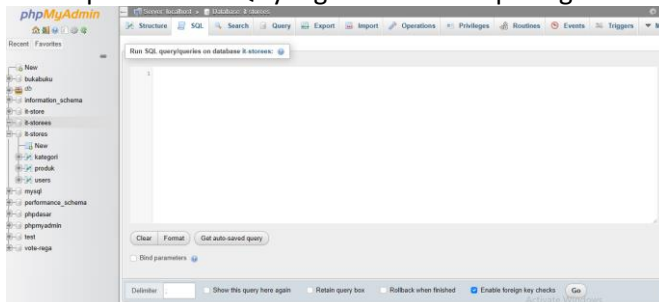
4. Pada bagian kiri anda akan melihat tampilan *database*, kemudian klik **NEW**.
5. Kemudian beri nama *database* dengan nama "it-store"

Databases



Gambar 32 Tampilan membuat nama *database*

6. Dan *database* anda sudah siap digunakan untuk kebutuhan aplikasi
7. Setelah itu klik nama *database* yang sudah anda buat tadi di atas.
8. Kemudian pilih menu SQL yang ada di atas seperti gambar di bawah ini.



Gambar 33 Tampilan SQL

9. Setelah anda klik menu SQL , silahkan anda buat kode/ sintaks SQLnya seperti di bawah ini

```
CREATE TABLE users(  
    id_user INT PRIMARY KEY,  
    username VARCHAR(255),  
    email VARCHAR(255),  
    password VARCHAR(255),  
    level VARCHAR(10)  
)
```

Sintaks di atas memiliki arti , membuat tabel yang bernama “users” kemudian memiliki beberapa kolom yaitu ada kolom “id_user” yang memiliki tipe data INT (*Integer*) atau bilangan bulat ,lalu kolom ini dijadikan “PRIMARY KEY” yang artinya datanya tidak boleh duplikat

atau biasa disebut dengan unik, kemudian ada kolom “username” yang memiliki tipe data “VARCHAR” yang artinya bisa di isi dengan huruf maupun angka yang dimana hanya bisa memuat “255” karakter saja. Lalu ada kolom “email” yang sama seperti username memiliki tipe data “VARCHAR” dengan “255” karakter saja. Di bawah “email” ada kolom “password” yang sama juga dengan “username” dan “email” dengan tipe data “VARCHAR” dengan maksimal karakter “255”. Dan yang terakhir ada kolom “level” sama juga seperti 3 kolom tadi namun hanya “10” karakter saja.

10. Kemudian langkah selanjutnya membuat tabel kategori seperti kode di bawah ini.

```
CREATE TABLE kategori(  
    id_kategori INT PRIMARY KEY,  
    id_user INT,  
    nama VARCHAR(100),  
    FOREIGN KEY(id_user) REFERENCES users(id_user) ON DELETE  
    RESTRICT  
)
```

Sintaks SQL di atas memiliki arti membuat tabel “kategori” dengan 3 kolom. Kolom pertama “id_kategori” sama seperti “id_user” yang sudah dijelaskan di atas tadi, kemudian ada kolom “id_user” yang mana nanti akan digunakan untuk keperluan relasi antara tabel “users” dan tabel “kategori”. Dan yang terakhir ada kolom “nama” yang bertipe data “VARCHAR” dengan maksimal karakter “100” karakter saja. Lalu ada kode “FOREIGN KEY” itu artinya kita menggunakan kolom “id_user” yang ada pada tabel “kategori” untuk di jadikan “FOREIGN KEY” yang nantinya akan menjadi penghubung antara tabel “users” dan “kategori”, lalu ada “REFERENCES” itu berarti kita mengambil kolom “id_user” dari tabel “users” yang di mana kolom tersebut menjadi penghubung. Terakhir ada “ON DELETE RESTRICT” itu memiliki arti jika ada yang ingin menghapus tabel ini , maka akan secara otomatis *tergenerate* oleh sistem *error*. Tabel ini digunakan untuk menampung data-data pengguna terutama penjual. Tabel ini digunakan untuk penyimpanan kategori-kategori yang diinputkan oleh pengguna.

11. Langkah terakhir yang kita buat di menu SQL ini membuat kode/sintaks sql dari tabel produk. Ikuti langkah-langkah berikut ini.

```
CREATE TABLE produk(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    id_kategori INT,  
    id_user INT,  
    nama_produk VARCHAR(255),  
    harga DOUBLE,  
    foto VARCHAR(100),  
    stok INT,  
    detail TEXT,  
    FOREIGN KEY(id_kategori) REFERENCES  
        kategori(id_kategori) ON DELETE RESTRICT,  
    FOREIGN KEY(id_users) REFERENCES users(id_user) ON  
        DELETE RESTRICT  
)
```

Kode program di atas adalah cara untuk membuat tabel “produk” yang diawali dengan kolom “id” (id produk), lalu ada kolom “id_kategori” yang digunakan untuk penghubung antara tabel “kategori” dan “produk”, kemudian ada kolom “id_user” sama fungsinya seperti kolom “id_kategori” untuk penghubung tabel, kemudian ada kolom “nama_produk” dengan tipe data “VARCHAR”, lalu ada kolom harga dengan tipe data “DOUBLE” yang artinya kita dapat menyimpan banyak sekali karakter dalam tipe data ini. Setelah itu ada pembuatan “FOREIGN KEY” sama seperti pada pembuatan tabel “kategori” yang di mana digunakan sebagai penghubung antara 2 tabel. Tabel “produk” ini digunakan untuk menyimpan produk-produk yang akan dimasukkan nanti.

Pada kode/sintaks sql yang telah dibuat di atas tadi, semua tabel memiliki *primary key* dan *foreign key* yaitu:

- Tabel “users” memiliki *primary key* pada kolom “id_user”
- Tabel “kategori” memiliki *primary key* pada kolom “id_kategori” serta memiliki *foreign key* yang terdapat pada kolom “id_user”

- Tabel “produk memiliki *primary key* yang terdapat pada kolom “id” dan juga memiliki 2 *foreign key* yaitu pada kolom “id_kategori” dan “id_user”.

Dengan terdapatnya *foreign key* dan *primary key* pada tabel-tabel yang ada, dapat memudahkan kita untuk membuat sistem yang lebih terstruktur dan mudah dikelola. Dari kode program / sintaks sql yang sudah dilakukan di atas sekarang dapat disimpulkan sebagai berikut:

- Tabel “users” memiliki hubungan/relasi *one-to-many* dengan tabel “kategori” dan “produk”, yang artinya satu user/pengguna dapat melakukan banyak penambahan data produk dan kategori. Kolom “id_user” pada tabel “kategori” dan “id_user” pada tabel “produk” digunakan sebagai *foreign key* untuk mengaitkan data dari ketiga tabel tersebut.
- Tabel “kategori” memiliki hubungan/relasi *one-to-many* dengan tabel “produk”, yang artinya satu kategori dapat memiliki banyak produk. Kolom “id_kategori” pada tabel “kategori” digunakan sebagai *foreign key* untuk mengaitkan data dari 2 tabel tersebut.

Database anda akan terlihat seperti berikut setelah data dari tabel yang berbeda dihubungkan dengan relasi ini, yang membuat system lebih terstruktur dan mudah dikelola.

Struktur *database*

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> kategori	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	32.0 K B	-
<input type="checkbox"/> produk	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	64.0 K B	-
<input type="checkbox"/> users	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 K B	-
3 tables	Sum	14	InnoDB	utf8mb4_general_ci	112.0 K B	0 B

Gambar 34 Tampilan struktur *database*

Tabel Kategori

		id_kategori	id_user	nama
<input type="checkbox"/>	Edit Copy Delete	1	1	Makanan
<input type="checkbox"/>	Edit Copy Delete	4	2	Minuman

Gambar 35 Tampilan tabel kategori

Tabel Produk

		id	id_kategori	id_user	nama_produk	harga	foto	stok	detail
<input type="checkbox"/>	 Edit  Copy  Delete	1	1	1	Gacoan	10000	647096f22e8a5.jpg	10	Makanan enak
<input type="checkbox"/>	 Edit  Copy  Delete	4	5	1	Kemeja	50000	6470a1f3a4acf.png	4	tttt

Gambar 36 Tampilan tabel produk

Tabel Users

		id_user	username	email	password	level
<input type="checkbox"/>	Edit Copy Delete	1	riza	riza@gmail.com	\$2y\$10\$mV7mJfAQ2ht216XRB3Hc6.hxEmqhQJf6HI60/WXgG5f...	penjual

Gambar 37 Tampilan tabel users

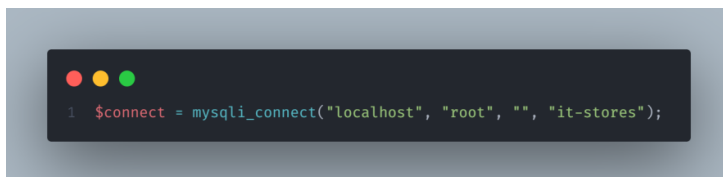
12. Itu dia tatacara perancangan *database* untuk membuat aplikasi ini. Selanjutn akan dilanjutkan dengan tutorial pembuatan aplikasi.

3.2.2 PEMBUATAN KODE LOGIN,REGISTER DAN LOGOUT

Dengan adanya perancangan *database* yang sudah dijelaskan sebelumnya kita dapat membuat kode program dengan cara terstruktur. Perlu diingat aplikasi ini dibuat dengan menggunakan bahasa pemograman PHP *Native* atau PHP *procedural*. Pertama yang akan kita buat adalah kode program login.

3.2.2.1 REGISTER

Pada aplikasi wirausaha ini terdapat proses *login* yang untuk mendapatkan akses sebagai penjual, maka dari itu untuk melakukan proses *register*. Silahkan anda buat file yang bernama "function.php". Berikut kode berserta penjelasannya.



Gambar 38 Kode koneksi ke *database*


```

1 function register($data)
2 {
3     global $connect;
4
5     $username = strtolower(stripslashes($data["username"]));
6     $email = strtolower(stripslashes($data["email"]));
7     $password = mysqli_real_escape_string($connect, $data["password"]);
8     $confirm = mysqli_real_escape_string($connect, $data["confirm"]);
9
10    $result = mysqli_query($connect, "SELECT username FROM users WHERE username = '$username'");
11
12    if (mysqli_fetch_assoc($result)) {
13        echo "<script>
14            alert('Username yang dipilih sudah terdaftar')
15        </script>";
16        return false;
17    }
18
19    if ($confirm !== $password) {
20        echo "<script>
21            alert('Password tidak sama')
22        </script>";
23        return false;
24    }
25
26    $hash = password_hash($password, PASSWORD_DEFAULT);
27
28    mysqli_query($connect, "INSERT INTO users VALUES('','$username','$email', '$hash', 'penjual')");
29
30    return mysqli_affected_rows($connect);
31 }

```

Gambar 39 Kode program untuk *login*

Pada gambar 3.10 terdapat source code untuk melakukan proses register. Pertama buat *function* dengan nama *register* kemudian diisi dengan paramater atau () yang isinya \$data yang berasal dari user yang menginputkan datanya lewat html di depan. Kemudian *variable* \$connect kita beri *global* agar dapat dipanggil karena *variable* \$connect terdapat pada luar lingkup / di luar *function*. Setelah itu buat *variable* untuk menampung data yang berasal dari *user* yang memasukan data lewat *form* yang dibuat menggunakan HTML, dengan setiap sintaks <input /> memiliki atribut "name" yang diisi untuk digunakan pada *function register*. Kemudian buat *variable* untuk menampung sintaks sql "SELECT * FROM users WHERE username = '\$username'". Arti dari sintaks tersebut adalah pilih semua dari tabel *users* berdasarkan kolom *username* yang *username*nya telah diinput oleh *user*. Kemudian buat kondisi apabila *username* sama dengan *username* yang sudah terdaftar pada database maka tidak bisa. Lalu ada kondisi lagi apabila *password* dan *confirm password* tidak sama maka *user* tidak bisa mendaftar. Kemudian buat *variable* untuk menampung *password* yang akan diacak menggunakan fungsi *password_hash()*, yang dimana memiliki dua parameter, yang pertama *variable* yang menampung *password* asli, dan yang kedua isi dengan algoritma yang ingin digunakan untuk mengacak *password*. Untuk lebih jelasnya silahkan untuk mencobanya.

Setelah membuat kode program pada file “function.php”, kita buat file dengan nama “register.php”. Ikuti kode di bawah ini.

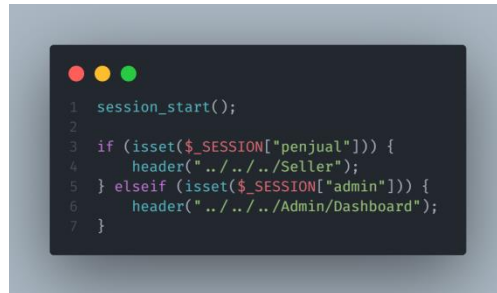


Gambar 40 Kode untuk mengecek *register*

Panggil file “function.php” dengan menggunakan sintaks “require ‘function.php’”. Lalu buat kondisi dengan “isset(\$_POST[‘submit’])” yang di mana “submit” itu diambil dari atribut *name* pada sintaks *button* di halaman register. Lalu buat kondisi lagi dengan melakukan kondisi dengan memanggil fungsi yang sudah dibuat pada halaman “function.php”, kemudian cek apakah pengguna benar-benar sudah melakukan *input* data mereka atau belum dengan menggunakan tanda lebih besar dari “>” 0. Jika ada maka akan mengeluarkan *alert* “Akun berhasil dibuat”.

3.2.2.2 LOGIN

Sesudah kita membuat program untuk melakukan pendaftaran akun, sekarang kita akan membuat kode program dari login. Silahkan buat file dengan nama “login.php” lihat kode program di bawah ini dan baca penjelasannya dengan cermat.

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The code is written in PHP and is as follows:

```
1 session_start();
2
3 if (isset($_SESSION["penjual"])) {
4     header("../..../Seller");
5 } elseif (isset($_SESSION["admin"])) {
6     header("../..../Admin/Dashboard");
7 }
```

Gambar 41 Kode program *session*.

Kode program di atas digunakan untuk melakukan *login filter*. Untuk melakukan *login filter* diawali dengan “*session_start()*”. Kemudian buat kondisi dengan buat dengan fungsi “*isset()*” yang di dalam *parameter* buat *variabel global* “*\$_SESSION[‘penjual’]*”, jika memenuhi maka akan dipindahkan ke halaman penjual. Jika “*\$_SESSION[‘admin’]*” maka akan dipindahkan ke halaman admin.

Kemudian panggil file “*function.php*” dengan menggunakan perintah *require* “*function.php*” seperti yang ada pada gambar di bawah ini.

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The code is written in PHP and is as follows:

```
1 require "../..../Function/index.php";
```

Gambar 42 Kode pemanggilan *file*

Setelah itu buat kode seperti yang ada pada bawah ini untuk melakukan proses login.

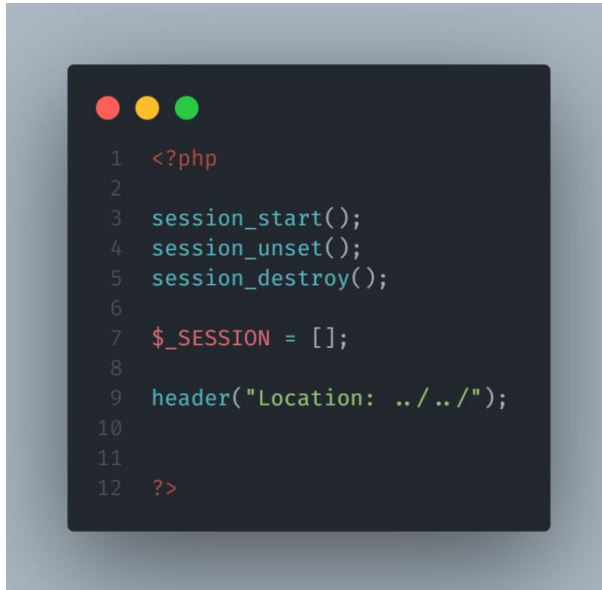
```
1 if (isset($_POST["submit"])) {
2
3     $username = $_POST["username"];
4     $password = $_POST["password"];
5
6     $result = mysqli_query($connect, "SELECT * FROM users WHERE username = '$username'");
7
8
9     //cek username
10    if (mysqli_num_rows($result) == 1) {
11
12        //cek password
13        $row = mysqli_fetch_assoc($result);
14        $verif = password_verify($password, $row["password"]);
15
16        if ($verif) {
17            $_SESSION["penjual"] = true;
18            $_SESSION["user"] = $username;
19            $_SESSION["id_user"] = $row["id_user"];
20            echo "<script>
21                window.location.href = '../..../Seller/'
22            </script>";
23            exit;
24        }
25    }
26    $cek = mysqli_num_rows($result);
27
28    if ($cek > 0) {
29
30        $data = mysqli_fetch_assoc($result);
31
32        if ($data["level"] == "admin") {
33            $_SESSION["admin"] = true;
34            echo "<script>
35                window.location.href = '../..../Admin/Dashboard'
36            </script>";
37        }
38    }
39    $error = true;
40 }
```

Gambar 43 Kode program untuk login.

Penjelasan kode dari gambar 3.14 dapat dilihat bahwa ada kondisi dimana apabila *button* yang atributnya *submit* maka akan dilakukan proses pembacaan kode dari *variable* “\$username” yang menampung data yang di *inputkan* oleh *user* dan juga *variable* “\$password”. Setelah itu buat *query* dengan menggunakan fungsi “*mysqli_query()*” yang di dalamnya di isi dengan sintaks sql “*SELECT * FROM users WHERE username = '\$username'*” yang ditampung pada *variable* \$result . Artinya pilih semua dari tabel *users* dimana kolom *username* sama dengan *username* yang di *inputkan* oleh *user*. Lalu ada kondisi lagi dimana ketika *username* yang di *inputkan* oleh *user* itu ada atau == 1 , maka cek lagi apakah *password* yang dimasukan sama atau tidak , jika sama maka buat “\$_SESSION[‘penjual’]” untuk melakukan login filter, “\$_SESSION[‘user’]” untuk menampilkan nama berdasarkan *user* yang login, dan “\$_SESSION[‘id_user’]” untuk mengambil *id* berdasarkan *user* yang login.

3.2.2.3 LOGOUT

Sesudah melakukan pembuatan proses *login* dan *register* selanjutnya pembuatan kode *logout*. Pertama buat “a href” dengan merujuk ke file “logout.php”. Silahkan buat terlebih dahulu file “logout.php”, setelah itu ikuti kode di bawah ini.



```
1  <?php
2
3  session_start();
4  session_unset();
5  session_destroy();
6
7  $_SESSION = [];
8
9  header("Location: ../../");
10
11
12  ?>
```

Gambar 44 Kode *logout*

“session_start()” digunakan untuk memulai *session* lalu di ikuti dengan “session_unset()” yang digunakan untuk menghapus semua *session* yang ada, setelah itu ada “session_destroy()” sama dengan “session_unset()” digunakan untuk memastikan *session* benar-benar hancur . Lalu dilanjutkan dengan “\$_SESSION” dengan isi kosong. Setelah itu arahkan ke halaman terdepan pada halaman web anda.

3.2.3 PEMBUATAN CRUD(CREATE,READ,UPDATE,DELETE)

Dalam aplikasi wirausaha ini pasti tidak akan lupa dengan CRUD ini. *Create* yang artinya membuat atau biasa disebut orang-orang dengan menambahkan data ke dalam *database*. *Read* artinya membaca maksudnya

menampilkan data yang ada di dalam *database* ke dalam tampilan aplikasi. *Update* berarti merubah artinya merubah data yang sudah ada pada *database* atau memperbarui data yang sudah ada. *Delete* berarti hapus artinya menghapus data yang ada pada *database*. Kita akan membahas cara untuk melakukan CRUD di dalam aplikasi wirausaha kita ini.

3.2.3.1 CREATE

Dalam tahap *create* ini kita akan membuat kode untuk melakukan penambahan data berdasarkan *user* yang melakukan *login* ada *create* kategori dan *create* produk. Perlu diingat bahwa untuk melakukan *create* harus menggunakan atribut “method = post” pada *form* HTML. Silahkan anda buat terlebih dahulu *file* dengan nama “create_kategori.php”, setelah anda membuatnya silahkan ikuti kode dibawah ini.

A screenshot of a code editor with a dark background and light-colored text. The code is a PHP function named 'tambah' that takes '\$data' as a parameter. It connects to a MySQL database at 'localhost' with 'root' user and an empty password for a database named 'it-store'. It then retrieves the user ID from the session ('\$_SESSION['id_user']') and the name from the input data (\$data['nama']). It constructs an SQL INSERT query to add a new category to the 'kategori' table, using the user ID and the name. Finally, it executes the query and returns the number of affected rows.

```
1 function tambah($data)
2 {
3     $connect = mysqli_connect("localhost","root","", "it-store");
4
5
6     $id_penjual = $_SESSION["id_user"] ;
7     $nama = $data["nama"];
8
9
10    $query = "INSERT INTO kategori VALUES ('','$id_penjual' , '$nama')";
11    mysqli_query($connect, $query);
12
13    return mysqli_affected_rows($connect);
14 }
```

Gambar 45 Kode fungsi tambah

Buatlah fungsi yang bernama “tambah” dengan parameter “\$data” , kemudian buat *variable* untuk menampung kode berisi koneksi ke dalam *database*, setelah itu buat *variable* dengan isi “\$_SESSION[‘id_user’]” yang di peroleh dari *user* yang login. Lalu buat *variable* untuk menampung data yang di inputkan oleh *user* yang berasal dari atribut name pada sintaks html “<input/>”. Setelah semuanya terbuat, buatlah sintaks sql “INSERT INTO kategori VALUES (‘ ’, ‘\$id_user’, ‘\$nama’)” maksud dari sintaks sql tersebut adalah memasukan data ke dalam tabel kategori dengan isi ‘ ’ (maksudnya

`id_kategori` tidak di isi karena kita sudah menyetel pada tabel kategori bahwa `id_kategori` *auto increment* yang artinya menginput otomatis oleh sistem), '`$id_user`', '`$nama`' simpan di *variable* `$query`, kemudian ambil data dengan fungsi "`mysqli_query()`".

Kemudian buat lagi kode program seperti di bawah ini untuk mengecek apakah *button* tambah sudah diklik atau belum. Silahkan ikuti kode program ini.

A screenshot of a code editor with a dark background and light-colored text. The code is PHP and is enclosed in a light gray border. It checks if the 'submit' button was clicked and if the 'tambah' variable is greater than 0. If both conditions are met, it shows an alert 'Data berhasil ditambahkan' and redirects to 'index_kategori.php'. Otherwise, it shows an alert 'Data gagal ditambahkan :('.

```
1  if (isset($_POST["submit"])) {  
2  
3      if (tambah($_POST) > 0) {  
4          echo  
5              "<script>  
6                  alert('Data berhasil ditambahkan');  
7                  window.location.href = 'index_kategori.php';  
8              </script>";  
9      } else {  
10         echo  
11             "<script>  
12                 alert('Data gagal ditambahkan :(');  
13             </script>";  
14         }  
15     }  
16 }
```

Gambar 46 Kode *button* diklik

Buat kondisi dengan jika tombol yang atribut *name* pada *button* bernama "submit" maka buat kondisi lagi dimana jika fungsi tambah lebih dari 0 maka keluarkan alert "data berhasil ditambahkan" dan pindahkan ke halaman "`index_kategori.php`" yang di mana itu tempat untuk menampilkan data kategori yang akan di bahas nanti.

Setelah membuat kode untuk kategori, sekarang kita akan membuat kode untuk menambahkan produk berdasarkan *user* yang *login*. Silahkan buat terlebih dahulu *file* "`create_produk.php`", kemudian lihat kode program di bawah ini.



Gambar 47 Kode untuk *create* produk

Buat fungsi bernama “tambahProduk” dengan isi di dalamnya *variable* untuk menampung koneksi ke *database* dilanjutkan dengan *variable* untuk menampung *id user* yang sedang *login*, perlu diingat bahwa untuk melakukan pemanggilan “\$_SESSION” harus diawali dengan “session_start()” pada awal kode program, kemudian buat *variable* “\$produk” dengan isi data dari atribut name pada *input* , lalu lakukan seterusnya sampai dengan akhir kolom dari tabel kategori. Untuk bagian foto ini spesial karena tidak seperti kode create sebelumnya. Untuk pembahasannya akan dibahas setelah ini. Lalu buat sintaks sql untuk memasukan data ke *database*.

Kemudian buat fungsi *upload* untuk menambahkan foto ke dalam *database*. Silahkan anda ikuti kode program di bawah ini. Sebelum kalian membuat fungsinya, kalian pastikan dulu di bagian form untuk menambahkan atribut *enctype* pada bagian *form*.



Gambar 48 Atribut *enctype*


```

1 function upload()
2 {
3     $namaFile = $_FILES["foto"]["name"];
4     $error = $_FILES["foto"]["error"];
5     $tmpName = $_FILES["foto"]["tmp_name"];
6
7     //cek apakah ada foto yg di upload atau tidak
8     if ($error == 4) {
9         echo
10         "<script>
11             alert('Upload foto terlebih dahulu');
12         </script>";
13         return false;
14     }
15
16     $ekstensiFotoValid = ["jpg", "jpeg", "png"];
17     $ekstensiFoto = explode('.', $namaFile);
18     $ekstensiFoto = strtolower(end($ekstensiFoto));
19
20     if(!in_array($ekstensiFoto, $ekstensiFotoValid)){
21         echo
22         "<script>
23             alert('Upload foto dongg');
24         </script>";
25         return false;
26     }
27
28     //generate nama foto baru
29
30     $namaFileBaru = uniqid();
31     $namaFileBaru .= '.';
32     $namaFileBaru .= $ekstensiFoto;
33
34
35
36
37     move_uploaded_file($tmpName, 'img/'.$namaFileBaru);
38
39     return $namaFileBaru;
40 }

```

Gambar 49 Kode untuk fungsi *upload*

Fungsi *upload* ini berisi *variable global* “\$_FILES” yang isi pertamanya “[‘foto’]” diambil berdasarkan atribut *name* pada *input* di HTML, lalu isi kedua “[‘name’]” itu bawaan dari atribut *enctype* tadi di atas dan seterusnya. Kemudian cek apakah foto yang di *upload error* atau tidak, *error* pada foto ini memang 4 tidak bisa di ubah lagi, jika kurang berarti tidak *error*, jika *error* maka keluarkan *alert* seperti yang ada pada kode di atas. Setelah itu buat *variable* yang berisi nama ekstensi apa saja yang boleh digunakan, lalu ada *variable* untuk menambahkan “.” setelah nama foto. Lalu buat kondisi dengan jika tidak ada foto yang dimasukan , maka keluarkan *alert*. Setelah semua sudah dibuat sekarang buat *variable* untuk menampung nama foto baru agar tidak terjadi duplikat dengan cara menambahkan “uniqid()” . Terakhir pindahkan foto ke *folder* yang anda buat seperti contoh kode di atas pada *folder* “img/”.

Jangan lupa juga buat kondisi pengecekan *button* diklik atau belum seperti kategori tadi.



Gambar 50 Kode pengecekan tombol

3.2.3.2 READ

Setelah kita sudah membuat kode program untuk melakukan proses *login*, kali ini kita akan membuat cara agar data yang ada di dalam database dapat ditampilkan pada halaman *website* khusus penjual seperti *admin* namun bukan *admin* dari *website*, tetapi admin penjual atau wirausahawan. Pertama buat *file* yang bernama “*index_kategori.php*” , setelah itu panggil data produk di dalam *database* kategori. Untuk lebih lanjut, silahkan ikuti kode program di bawah ini.



Gambar 51 Sintaks SQL untuk program read

Langkah awal kita panggil *file* “*function.php*” untuk memanggil koneksi *database*. Setelah itu untuk melakukan pemanggilan “*id_user*” berdasarkan *user* yang *login*, kita memerlukan *session* yang mana harus di awali dengan “*session_start*”. Kemudian buat *variable* yang menampung koneksi *database*

dengan sintaks SQL . Sintaks SQL di atas berarti pilih semua dari tabel kategori dimana kolom “id_user” sama dengan user yang login.

Setelah mengikuti kode program di atas , langkah selanjutnya membuat tabel pada file “index_kategori.php”. Perlu diingat pastikan , kode program di atas sudah dibuat. Silahkan ikuti kode program di bawah ini.

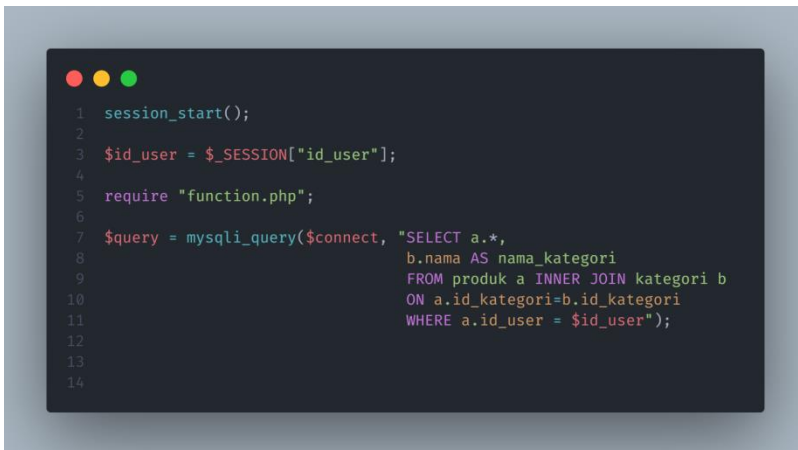
A screenshot of a code editor with a dark background and light-colored text. The code is PHP, showing a table structure with three columns: 'No.', 'Kategori', and 'Aksi'. It uses a while loop to fetch data from a database and display it in the table. The 'Aksi' column contains a button that links to a detail page for each category.

```
1 <table class="table ms-0">
2   <tr>
3     <th scope="col">No.</th>
4     <th scope="col">Kategori</th>
5     <th scope="col">Aksi</th>
6   </tr>
7   <?php $i = 1; ?>
8   <?php while ($row = mysqli_fetch_assoc($queryKategori)) : ?>
9     <tr>
10      <td scope="row"><?=$i; ?></td>
11      <td><?=$row["nama"]; ?></td>
12      <td>
13        <a class="btn btn-secondary"
14          href="detail_index_kategori.php?id=
15            <?=$row["id_kategori"]; ?>">
16          Cari
17        </a>
18      </td>
19    </tr>
20    <?php $i++; ?>
21  <?php endwhile; ?>
22 </table>
```

Gambar 52 Kode untuk menampilkan data kategori

Buatlah tabel pada form HTML anda yang berisi no,kategori,dan aksi. Kemudian buka sintaks PHP dengan value “\$i = 1” yang digunakan untuk penomoran tabel, lalu buat perulangan while untuk menampilkan data seperti kode di atas. Lalu pada kolom pertama panggil “\$i” untuk menampilkan nomor, kemudian panggil data yang ada di database “\$row” di ambil dari perulangan while, sedangkan [“nama”] diambil dari nama kolom di tabel kategori.

Setelah kita menampilkan data kategori ke file “index_kategori.php” , kita akan menampilkan data produk pada file “index_produk.php”. Silahkan buat terlebih dahulu file “index_produk.php”, kemudian ikuti kode di bawah ini.



```
1 session_start();
2
3 $id_user = $_SESSION["id_user"];
4
5 require "function.php";
6
7 $query = mysqli_query($connect, "SELECT a.*,
8                                b.nama AS nama_kategori
9                                FROM produk a INNER JOIN kategori b
10                               ON a.id_kategori=b.id_kategori
11                               WHERE a.id_user = $id_user");
12
13
14
```

Gambar 53 Sintaks SQL untuk mengambil data produk

Awali dengan “session_start()” untuk memulai *session* , lalu buat lah *variable* untuk menampung “id_user” berdasarkan *user* yang *login* . Setelah itu panggil *file* “function.php” untuk menggunakan *variable* “\$connect” . Kemudian buat *variable* untuk menampung *query* . Penjelasan *query* di atas adalah pilih semua dari tabel a yang berarti tabel produk lalu pilih tabel b kolom nama di ganti dengan nama “nama_kategori” dari tabel produk yang di inisialiasi dengan a dan ambil bagian tengah saja yaitu bagian yang menjadi *foreign key* pada tabel produk ke tabel kategori yang di inialisasi dengan b dengan kolom “id_kategori “ di tabel a sama dengan “id_kategori” yang ada di kolom b, dimana “id_user” pada tabel produk sama dengan “id_user” berdasarkan *user* yang *login*.

Setelah anda membuat kode untuk memanggil data yang ada di tabel produk, langkah selanjutnya buat tabel seperti pada pemanggilan data produk.Silahkan lanjutkan kode program pada *file* “index_produk.php”. Ikuti kode program di bawah ini.



Gambar 54 Kode untuk menampilkan data produk

Buatlah tabel pada *form* HTML anda yang berisi no,kategori,dan aksi. Kemudian buka sintaks PHP dengan *value* “\$i = 1” yang digunakan untuk penomoran tabel, lalu buat perulangan *while* untuk menampilkan data seperti kode di atas. Lalu pada kolom pertama panggil “\$i” untuk menampilkan nomor, kemudian panggil data yang ada di *database* , gunakan *variable* “\$produk” untuk menampung data yang di ambil dari perulangan while, sedangkan [“foto”] diambil dari nama kolom di tabel kategori dan seterusnya sampai bagian [“detail”].

3.2.3.3 UPDATE

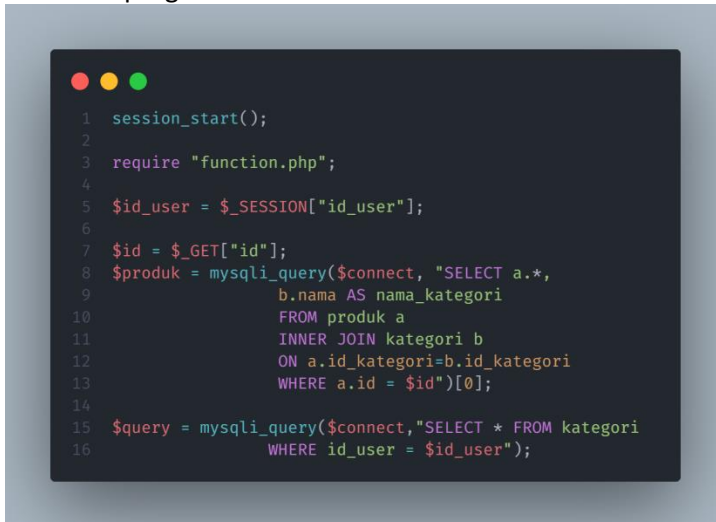
Pada tahap *update* ini yang anda perlukan pertama kali buat *tag* HTML “” yang href='update_produk.php' “ nya merujuk pada *file* “update.php” kemudian buat *get* dengan cara seperti yang ada di bawah ini. Silahkan buat *file* “update.php” dan ikuti kode di bawah ini.



Gambar 55 Kode untuk menggunakan *get*

Arahkan ke file “update.php” dengan “?” yang berarti memanggil *method get* lalu beri tanda sesuka hati anda namun untuk memudahkan beri tanda dengan nama “id” yang isinya *id* dari tabel produk. *Method get* digunakan untuk menampilkan data melalui *url*.

Setelah mengarahkan ke file “update_produk.php”, lalu kita buat program untuk mengambil data dari tabel produk berdasarkan *user* yang *login*. Silahkan ikuti kode program di bawah ini.



Gambar 56 Kode *query* ambil data

Langkah awal sama seperti sebelum-sebelumnya , kemudian buat *variable* untuk menampung *method get* . `$_GET["id"]` diambil berdasarkan *url* yang dikirim setelah mengklik *update*(a href tadi). Lalu ambil data dari tabel produk , penjelasan kodenya sama seperti pada pembahasan *read* sebelumnya. Langkah selanjutnya mengambil data katageri berdasarkan *user* yang *login*.

Sebelum kita lanjut ke pembuatan fungsi ubah, kita buat agar data yang sudah ada tampil dan *user* akan mengubah data jika ada kesalahan pada proses penambahan data. Silahkan ikuti kode di bawah ini, perlu di ingat kode program ini tersimpan di *file* yang sama “update_produk.php”.

A screenshot of a code editor with a dark background and light-colored text. The code is a mix of PHP and HTML, used for updating product data. It includes two hidden input fields for 'fotoLama' and 'id', and a text input field for 'nama_produk'. The code is numbered from 1 to 24. The PHP code uses \$produk array to retrieve and set values. The HTML code uses Bootstrap classes for styling the form elements.

```
1 <table>
2   <input
3     type="hidden"
4     name="fotoLama"
5     value="<?= $produk["foto"]; ?>"
6   <input
7     type="hidden"
8     name="id"
9     value="<?= $produk["id"] ?>"
10  <tr>
11    <td class="pe-4 pb-4">
12      <label for="produk">Nama Produk</label>
13    </td>
14    <td class="pe-3 pb-4"></td>
15    <td class="pb-4">
16      <input
17        type="text"
18        placeholder="isi produk ..."
19        name="produk" id="produk"
20        required
21        class="form-control"
22        value="<?= $produk["nama_produk"]; ?>"
23      </td>
24    </tr>
```

Gambar 57 Kode menampilkan data untuk di ubah *part 1*

Kode di berikan terpisah agar terlihat jelas. Pada bagian pertama buat *tag input* dengan *type hidden* untuk kebutuhan penambahan data saat di ubah lalu isi dengan foto yang lama, lalu *tag input* kedua isi dengan *id* dari tabel produk. Kemudian untuk melakukan perubahan data kita perlu atribut *value* yang digunakan untuk menampilkan data sebelumnya. Pada bagian pertama isi dengan nama dari produknya [“nama_produk”] sesuaikan dengan yg ada pada bagian tabel produknya.

```
1 <tr>
2   <td class="pe-4 pb-4">
3     <label for="kategori">Kategori</label>
4   </td>
5   <td class="pe-3 pb-4"></td>
6   <td class="pb-4">
7     <select
8       class="form-select"
9       aria-label="Default select example"
10      name="kategori"
11      id="kategori">
12       <?php foreach ($query as $barang) : ?>
13         <option value="<?= $barang["id_kategori"]; ?>">
14           <?= $barang["nama"]; ?>
15         </option>
16       <?php endforeach; ?>
17     </select>
18   </td>
19 </tr>
```

Gambar 58 Kode menampilkan data untuk di ubah *part 2*

Setelah kita menampilkan nama produk yang sudah ada pada tabel, sekatang buat untuk menampilkan data kategori berdasarkan *user* yang login dengan cara menggunakan *tag select* yang berisi *option* dengan *value* "id_kategori" yang ada pada tabel produk, lalu tampilkan nama kategori.

```
1 <tr>
2   <td class="pe-4 pb-4">
3     <label for="foto">Foto</label>
4   </td>
5   <td class="pe-3 pb-4"></td>
6   <td class="pb-4">
7     <input
8       type="file"
9       name="foto"
10      id="foto"
11      class="form-control">
12   </td>
13   <td class="pb-4 ps-2">
14     "
16       alt="foto produk"
17       width="100">
18   </td>
19 </tr>
```

Gambar 59 Kode menampilkan data untuk di ubah *part 3*

Kemudian buat *tag input type file* untuk mengubah foto jika foto yang lama tidak sesuai dengan produk. Jika masih sesuai maka tidak perlu di isi. Setelah itu tampilkan foto sebelumnya agar *user* tau foto apa yang sudah ada pada produk ini, jika sudah sesuai maka tidak perlu untuk merubahnya.

```
1  <tr>
2    <td class="pe-4 pb-4">
3      <label for="stok">Ketersediaan Stok</label>
4    </td>
5    <td class="pe-3 pb-4">:</td>
6    <td class="pb-4">
7      <input
8        type="number"
9        placeholder="isi stok..."
10       name="stok" id="stok"
11       required
12       class="form-control"
13       value="<?= $produk["stok"]; ?>"
14     </td>
15  </tr>
16  <tr>
17    <td class="pe-4 pb-4">
18      <label for="detail">Detail</label>
19    </td>
20    <td class="pe-3 pb-4">:</td>
21    <td class="pb-4">
22      <textarea
23        name="detail"
24        id="detail"
25        cols="30"
26        rows="10"
27        class="form-control"
28        required>
29        <?= $produk["detail"]; ?>
30      </textarea>
31    </td>
32  </tr>
```

Gambar 60 Kode menampilkan data untuk di ubah *part 4*

Setelah menampilkan foto , sekarang bagian untuk menampilkan ketersediaan stok, sama seperti sebelumnya tampilkan berdasarkan nama

kolom yang ada pada tabel produk dan untuk detail juga sama seperti sebelum-sebelumnya.



Gambar 61 Kode untuk bagian *button*

Terakhir buat *tag button* untuk mengubah data yang salah. Berikan atribut *name submit* yang akan digunakan pada bagian *backend*.Dilanjutkan dengan pembuatan fungsi untuk merubah data. Silahkan kode di bawah ini anda simpan di bawah kode yang ada pada gambar 3.27 .



Gambar 62 Kode untuk mengambil data dari *user*

Pada kode program di atas dapat dilihat ada fungsi *ubah* yang di dalamnya terdapat beberapa *variable* yang menampung data yang dikirim dari *user* berdasarkan atribut *tag input name* , untuk bagian foto maksudnya itu jika foto yang di *upload* tidak ada maka gunakan foto sebelumnya, jika ada yang di *upload* maka gunakan fungsi “*upload()*” yang sudah dibuat pada bagian *create*, *copy* kan dan *paste* ke bagian bawah fungsi *upload* .

Setelah mendapatkan data dari *user* kita sekarang buat *query* untuk mengubah data.Silahkan ikuti kode program di bawah ini.



```
1  $query = "UPDATE produk SET
2          id_kategori = '$kategori',
3          nama_produk = '$produk',
4          harga      = '$harga',
5          foto       = '$foto',
6          stok       = '$stok',
7          detail     = '$detail'
8
9
10         WHERE id = $id
11     ";
12
13     mysqli_query($connect, $query);
14
15     return mysqli_affected_rows($connect);
16 }
```

Gambar 63 Kode untuk mengubah data yang ada di *database*

Buat *variable* untuk menampung sintaks SQL yang ada pada di gambar 3.34. Dari *query* di atas bahwa dapat di artikan sebagai berikut ubah tabel produk dengan “id_kategori” sama dengan data yang dikirim oleh *user* dan seterusnya sampai “detail” dimana “id” (id produk) sama dengan id yang di dapat dari *url* .

Setelah semua sudah terbuat , langkah selanjutnya membuat kondisi apabila tombol sudah diklik atau belum. Silahkan ikuti kode program di bawah ini



Gambar 64 Kode untuk mengecek tombol

Untuk penjelasan kode program ini kurang lebih sama seperti yang ada pada bagian *create*. Silahkan dilihat kembali.

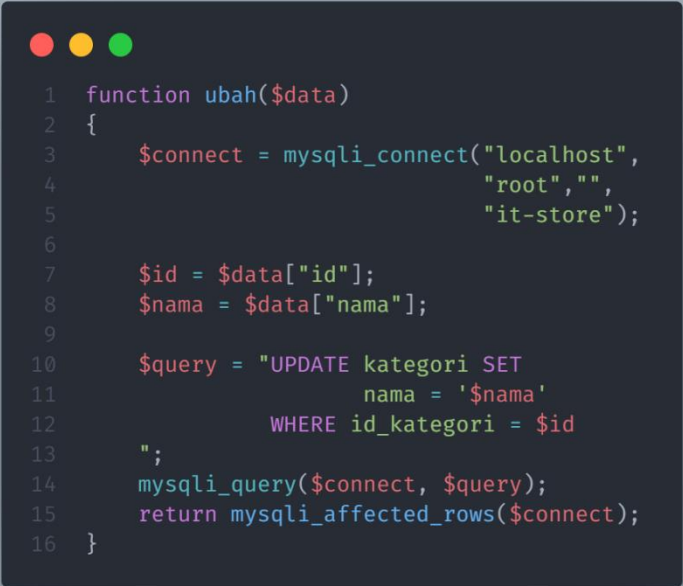
Setelah melakukan ubah data produk, sekarang melakukan ubah data kategori. Silahkan buat terlebih dahulu file “update_kategori.php”, kemudian ikuti kode program di bawah ini.



Gambar 65 Kode *query* data kategori

Penjelasan dari kode di atas kurang lebih sama seperti ubah data produk, yang berbeda hanya pada pengambilan data. Maksud dari *query* di atas adalah pilih semua dari tabel kategori dimana “id_kategori” sama dengan id yang di ambil dari *url*.

Setelah membuat *query* data kategori selanjutnya membuat fungsi ubah untuk kategori. Silahkan ikuti kode program di bawah ini.



```

1  function ubah($data)
2  {
3      $connect = mysqli_connect("localhost",
4                                "root", "",
5                                "it-store");
6
7      $id = $data["id"];
8      $nama = $data["nama"];
9
10     $query = "UPDATE kategori SET
11                nama = '$nama'
12                WHERE id_kategori = $id
13            ";
14     mysqli_query($connect, $query);
15     return mysqli_affected_rows($connect);
16 }

```

Gambar 66 Kode untuk fungsi ubah kategori

Sama seperti pada ubah produk, gunakan *query* untuk mengubah data pada kategori berdasarkan id yang di ambil dari *url* .

Kemudian buat *tag form* untuk menampilkan data kategori yang sudah dimasukan oleh *user*. Silahkan ikuti kode di bawah ini.

```

1  <table>
2      <input
3          type="hidden"
4          name="id"
5          value="<?= $kategori["id_kategori"] ?>"
6      >
7      <tr>
8          <td class="p-5">
9              <label for="nama">
10                 Nama Kategori
11             </label>
12         </td>
13         <td class="pe-3"></td>
14         <td>
15             <input
16                 type="text"
17                 name="nama"
18                 value="<?= $kategori["nama"]; ?>"
19                 id="nama"
20                 require
21                 class="form-control"
22             </td>
23         </tr>
24         <tr>
25             <td></td>
26             <td></td>
27             <td>
28                 <button
29                     type="submit"
30                     name="submit"
31                     class="btn btn-primary">
32                     Ubah
33                 </button>
34             </td>
35         </tr>
36     </table>

```

Gambar 67 Kode untuk menampilkan data yang sudah di *input*

Buat *tag input* dengan *type hidden* untuk menampilkan id dari kategori, kemudian buat *tag input* dengan *value* data yang sudah ada pada database yang ingin di ubah

Lalu buat kondisi apakah tombol ubah sudah ditekan atau belum. Silahkan ikuti kode di bawah ini.

```

1  if (isset($_POST["submit"])) {
2
3      if (ubah($_POST) > 0) {
4          echo
5              "<script>
6                  alert('Data berhasil diubah');
7                  window.location.href = 'index_kategori.php';
8              </script>";
9      } else {
10         echo
11             "<script>
12                 alert('Data gagal diubah :( ');
13             </script>";
14     }
15 }

```

Gambar 68 Kode untuk kondisi tombol

Buat kondisi dengan ["submit"] sesuai dengan *name* pada atribut *button*. Jika berhasil maka mengeluarkan pesan "Data berhasil di ubah", jika gagal maka akan mengeluarkan pesan "Data gagal di ubah".

3.2.3.4 DELETE

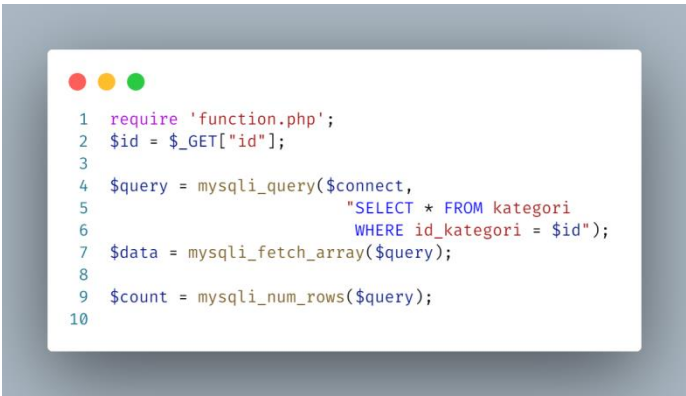
Sebelum anda membuat fungsi hapus, buat *tag a href* yang menuju ke *file* "delete.php". Silahkan buat terlebih dahulu *file* "delete_kategori.php" kemudian ikuti kode program di bawah ini yang di simpan pada *file* "index_kategori.php".



```
1 <a
2 href="delete.php?id=<?= $data["id_kategori"]; ?>"
3 class="btn btn-danger"
4 onclick="return confirm('Yakin mau di hapus?')">
5 Hapus
6 </a>
```

Gambar 69 Kode untuk mengarahkan ke *file*

Arah kan ke *file* "delete_kategori.php" dan gunakan *method get* untuk mengambil data "id_kategori" dari *url*. Setelah membuat kode program di atas , silahkan buat fungsi hapus pada *file* "delete_kategori.php". Ikuti kode program di bawah ini.



```
1 require 'function.php';
2 $id = $_GET["id"];
3
4 $query = mysqli_query($connect,
5                          "SELECT * FROM kategori
6                          WHERE id_kategori = $id");
7 $data = mysqli_fetch_array($query);
8
9 $count = mysqli_num_rows($query);
10
```

Gambar 70 Kode *query* untuk mengambil data

Ambil data dari *url* menggunakan “\$_GET” , kemudian buat *variable* yang menampung *query* untuk mengambil data, arti dari sitaks *query* di atas yaitu, ambil semua dari tabel kategori dimana “id_kategori” sama dengan id yang diambil dari *url*. Kemudian buat *variable* yang menampung *variable* yang di gunakan untuk menghitung jumlah yang ada pada tabel kategori “mysql_num_rows()”.

```
1  if ($count > 0) {
2      echo
3      "<script>
4          alert('Kategori tidak dapat di hapus karena sudah memiliki produk')
5          window.location.href = 'index_kategori.php';
6      </script>";
7      die();
8  }
9
10 function hapus($id) {
11     $connect = mysqli_connect(
12         "localhost",
13         "root",
14         "",
15         "it_store");
16     mysqli_query($connect, "DELETE FROM kategori WHERE id_kategori = $id");
17     return mysqli_affected_rows($connect);
18 }
19 }
```

Gambar 71 Kode fungsi hapus

Kemudian buat kondisi jika lebih data yang ada di tabel kategori lebih dari 0 maka tidak bisa di hapus. Lalu buat fungsi untuk melakukan hapus data yg di dalamnya ada *query* untuk hapus data. Arti dari *query* di atas adalah hapus dari tabel kategori dimana “id_kategori” sama dengan id yang diambil dari *url*.

```
1  if ( hapus($id) > 0 ) {
2      echo
3      "<script>
4          alert('Data berhasil dihapus');
5          window.location.href = 'index_kategori.php';
6      </script>";
7  } else {
8      echo
9      "<script>
10         alert('Data gagal dihapus');
11         window.location.href = 'index_kategori.php';
12     </script>";
13 }
```

Gambar 72 Kode untuk kondisi hapus

Jika sudah buat kondisi dengan apabila data yang ada maka keluarkan pesan “Data berhasil di hapus” lalu arahkan ke *file* “index_kategori.php”, jika gagal akan mengeluarkan pesan “Data gagal di hapus”.

Setelah kita membuat fungsi untuk hapus kategori, selanjutnya membuat fungsi untuk hapus prouduk. Silahkan buat *file* “delete_produk.php” untuk menyimpan fungsi hapus produk. Pertama pada *file* “index_produk.php” buat *tag a href* yang merujuk pada *file* “delete_produk.php”. Silahkan ikuti kode program di bawah ini.

```
1 <a href="delete_produk.php?id=<?= $produk["id"]; ?>"
2   class="btn btn-danger"
3   onclick="return confirm('Yakin mau di hapus?')>Delete
4 </a>
```

Gambar 73 Kode untuk mengarahkan ke *file delete*


Arah kan ke *file* “delete_produk.php” dan gunakan *method get* untuk mengambil data “id” dari *url*. Setelah membuat kode program di atas , silahkan buat fungsi hapus pada *file* “delete_produk.php”. Ikuti kode program di bawah ini.

```
1 $id = $_GET["id"];
2
3 $query = mysqli_query($connect, "SELECT *
4                                   FROM produk
5                                   WHERE id = $id");
```

Gambar 74 Query untuk mengambil data

Ambil data dari *url* menggunakan “\$_GET” , kemudian buat *variable* yang menampung *query* untuk mengambil data, arti dari sitaks *query* di atas yaitu, ambil semua dari tabel produk dimana “id” sama dengan *id* yang di ambil dari *url* yang di dapatkan dari *method get*.

Lalu buat fungsi untuk menghapus data produk. Silahkan ikuti kode program di bawah ini



```
1 function hapus($id) {  
2     $connect = mysqli_connect("localhost",  
3         "root",  
4         "",  
5         "it-store");  
6     mysqli_query($connect, "DELETE FROM produk  
7         WHERE id = $id");  
8  
9     return mysqli_affected_rows($connect);  
10 }
```

Gambar 75 Kode fungsi hapus produk

Lalu buat kondisi untuk mengecek apakah tombol sudah ditekan atau belum. Silahkan lihat kode program di bawah ini.



```
1 if ( hapus($id) > 0 ) {  
2     echo  
3     "<script>  
4         alert('Data berhasil dihapus');  
5         window.location.href = '../';  
6     </script>";  
7 } else {  
8     echo  
9     "<script>  
10         alert('Data gagal dihapus');  
11         window.location.href = '/';  
12     </script>";  
13 }
```

Gambar 76 Kode untuk kondisi tombol hapus produk

Untuk penjelasan kode di atas kurang lebih sama dengan kategori.

BAB 4

PENUTUP

Selama penulisan buku ini, pembuatan sistem kewirausahaan mahasiswa Universitas Logistik & Bisnis Internasional dibahas. Proses membangun sistem ini melibatkan beberapa langkah, dimulai dengan mendesain dan membuat rancangan *database*, kemudian dilanjutkan dengan membangun sistem dari aplikasi, menggunakan menggunakan bahasa pemrograman PHP, serta menginstal Visual Studio Code dan XAMPP.

Dengan mengembangkan sistem kewirausahaan ini, diharapkan dapat meningkatkan pengembangan ekonomi kreatif dan mendorong mahasiswa Universitas Logistik & Bisnis Internasional untuk aktif berwirausaha dan mempercepat proses ekonomi berbagai sektor di kampus Universitas Logistik & Bisnis Internasional. Oleh karena itu, diharapkan sistem kewirausahaan mahasiswa Universitas Logistik & Bisnis Internasional akan memberikan solusi yang lebih baik dan lebih efisien di bidang layanan penjual dan pelanggan.

DAFTAR PUSTAKA

Devy Kusumaningrum ,Karjono (2020). Majalah Ilmiah Gema Maritim Vol 22 No 2 tahun 2020(165-166).

<https://kamus.tokopedia.com/b/bahasa-program/>.

Gloldbot, Micah. 2016. *“Frontend Architecture for Design System”* . California: O'Reilly Media.

Gumolung S. G. Mulia – Analisa Hyper Text Markup Language (HTML) Versi 5.

Duckett, J. (2011). *HTML & CSS: Design and Build Web Sites*. . Germany: : Wiley.

Selli Mariko, No 1, April 2019 (80-91) APLIKASI WEBSITE BERBASIS HTML DAN JAVASCRIPT UNTUK MENYELESAIKAN FUNGSI INTEGRAL PADA MATA KULIAH KALKULUS Volume 6

Rozi A. Zaenal & Comunity Smit Dev, 2015 Bootstrap Design Framework, Jakarta: PT Elex Media Komputindo.

Muhammad Saed Novendri , Ade Saputra , Chandra Eri Firman, APLIKASI INVENTARIS BARANG PADA MTS NURUL ISLAM DUMAI MENGGUNAKAN PHP DAN MYSQL vol 10 nomor 2 Mei 2019

GLOSARIUM

A

Admin : Kependekan dari kata administrasi yang merupakan pengelola atau pemilik.

B

Back-and : Bagian dari situs *web* yang tidak dilihat oleh pelanggan.

C

Client-side : Sisi pengguna dan *server-side* atau sisi penyedia.

D

Developer : Seseorang yang memiliki tugas sebagai perancang software, *web* atau aplikasi.

E

F

Framework : kKerangka kerja yang digunakan untuk mengembangkan *web*.

Front-end : Bagian *web* yang menampilkan tampilan para pengguna.

G

GNU : Sistem operasi komputer yang terdiri dari perangkat lunak bebas.

H

I

Inovatif : Sesuatu yang bersifat memperkenalkan sesuatu yang baru atau bersifat pembaruan.

Investor : Orang atau lembaga yang melakukan investasi dalam suatu hal dengan tujuan untuk membuat keuntungan finansial

J

Jasa : Perbuatan yang memberikan segala sesuatu yang diperlukan orang lain.

K

L

Login : Istilah keamanan pada sistem komputer, berupa proses masuk bagi pengguna.

Logout: Istilah keamanan pada sistem komputer, berupa proses keluar bagi pengguna.

M

Mac Os : Sistem operasi yang menjalankan setiap Mac

N

NPM: Nomor pokok mahasiswa

O

Online : Interaksi yang dilakukan dalam jaringan.

P

Pengusaha: Orang yang berusaha dalam bidang perdagangan.

Primary key : Kode yang bersifat unik dan berbeda.

Produk : Barang atau jasa yang dibuat dan ditambah gunanya atau nilainya dalam proses produksi dan menjadi hasil akhir dari proses produksi itu.

Program : Rancangan yang akan dilaksanakan

Q

Query : Perintah yang digunakan untuk mendapatkan informasi dalam database dengan tujuan untuk melakukan tugas tertentu.

R

Register : Salah satu dari sekumpulan kecil tempat penyimpanan data yang merupakan bagian dari prosesor komputer.

S

Server : Sistem komputer yang menjalankan jenis layanan tertentu dalam sebuah jaringan komputer.

Sistem informasi : Seperangkat komponen yang saling berhubungan yang berfungsi untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi.

T

Teknologi : Metode ilmiah untuk mencapai tujuan praktis, ilmu pengetahuan terapan atau keseluruhan sarana untuk menyediakan barang-barang yang diperlukan bagi kelangsungan dan kenyamanan hidup manusia.

U

User : Pengguna.

V

W

Web : Sebuah sistem untuk mengakses, memanipulasi, dan mengunduh dokumen hipertaut yang terdapat dalam komputer yang dihubungkan melalui internet; jejaring; jaringan.

WebGL : Teknologi web yang menyuguhkan akselerasi grafis 3D ke dalam browser tanpa memasang perangkat lunak tambahan.

X

Y

Z

INDEKS

A

Admin 24, 25, 33, 40, 57

B

Back-end 5, 9, 10

C

Client-side 8

D

Developer 7, 9, 10

E

F

Framework 5, 9

Front-end 5, 6, 9

G

GNU 15

H

I

Inovatif 1, 3, 4

Investor 4

J

Jasa 1, 2, 4

K

L

Login 23, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46

Logout 30, 35

M

Mac Os 10

N

NPM 22

O

Online 6, 10

P

Pengusaha 2, 4

Primary Key 26, 27, 28

Produk 8, 9, 27, 28, 29, 36, 37, 38, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 55, 56

Program 5, 8, 15, 23, 28, 30, 31, 32, 33, 34, 37, 38, 40, 41, 42, 44, 45, 49, 50, 53, 55, 56, 58

Q

Query 34, 37, 42, 44, 49, 50, 51, 53, 54, 55

R

Register 21, 30, 31, 32, 35

S

Server 5, 8, 9, 10, 15, 20

T

Teknologi 6, 7

U

User 5, 22, 26, 27, 28, 29, 30, 31, 34, 36, 37, 38, 40, 41, 42, 44, 45, 46, 47, 48, 49, 51

V

W

Web 6, 7, 8, 10, 11, 15, 16, 35, 40

WebGL 7

X

Y

Z

TENTANG PENULIS




M. Fachriza Farhan lahir di Kabupaten Pekalongan pada tanggal 03 Mei 2004. Pendidikan tingkat dasar dan menengah atas ditempuh di Bandung, sedangkan pendidikan menengah pertama ditempuh di Pekalongan. Saat ini saya sedang menempuh perguruan tinggi jurusan DIV Teknik Informatika di Universitas Logistik dan Bisnis Internasional.



Muhammad Farhan Akbar Muhlis, lahir di Kota Parepare pada tanggal 24 Februari 2004. Pendidikan tingkat dasar hingga menengah ditempuh di Parepare. Menyelesaikan pendidikan dasar pada tahun 2016 di SD Negeri 38 Parepare selanjutnya menghabiskan kurang lebih tiga tahun di SMP Negeri 2 Rujukan Parepare, kemudian melanjutkan pendidikan di *boarding school* SMA Negeri 5 Parepare. Sekarang, tengah menempu studi sarjana terapan Teknik Informatika semester 2 di Universitas Logistik & Bisnis Internasional. SD hingga SMA aktif dalam organisasi kepramukaan, kesenian dan osis pada masa SMP-SMA. Saat ini di kampus ULBI aktif pada UKM Atmacipta Svara, UKM K-radio dan *Celebes Student Community*.



Noviana Riza, S.Si., M.T., SFPC, menempuh pendidikan sekolah tinggi pada Universitas Padjajaran kemudian melanjutkan S2 pada Institut Teknologi Bandung keduanya mengambil jurusan yang sama yaitu Teknik Informatika. Mendapat dua sertifikat lisensi internasional di antaranya *Microsoft Technology Associate* (MTA) AI-900 dan *Scrum Foundation Profesional Certificate* (SFPC). Saat ini aktif menjadi tenaga pengajar di Universitas Logistik & Bisnis Internasional



Di era globalisasi, tidak hanya perkembangan teknologi yang terus berkembang, tetapi semua bidang pekerjaan salah satunya dibidang kewirausahaan. Perkembangannya menuntut hadirnya teknologi bisnis sebagai alat penunjang untuk meningkatkan kinerja para pengusaha masa kini. Mengingat besarnya minat mahasiswa Universitas Logistik & Bisnis Internasional dalam berwirausaha khususnya di jurusan DIV Teknologi Informatika, maka penulis memandang perlu untuk menghadirkan suatu rancangan sistem yang dapat memudahkan kegiatan jual beli dan informasi tentang penjualan produk. Sehingga mahasiswa Universitas Logistik & Bisnis Internasional dapat mengelola usahanya secara maksimal.

Adanya sistem tersebut dapat membantu perkembangan kewirausahaan mahasiswa, memudahkan pembeli dalam melakukan pembelian dan memudahkan penjual dalam melakukan pencatatan pesanan. Dengan cara ini, penjual mempelajari jenis dan jumlah pesanan yang dilakukan oleh pembeli dan memfasilitasi proses pembelian pembeli. Sistem ini juga menyediakan kemampuan untuk memperluas menu atau sebaliknya, sehingga penjual tidak perlu khawatir untuk mengurangi atau menambah produk online di kemudian hari.

Tujuan penulisan buku ini adalah untuk menjelaskan cara membuat sistem kewirausahaan. Buku ini membahas tentang perancangan sistem informasi kewirausahaan untuk meningkatkan pengolahan data komoditas yang masuk dan keluar. Setiap bab yang disajikan dalam buku ini juga disertakan dalam kode program yang dapat diakses.

M. FACHRIZA FARHAN
MUHAMMAD FARHAN AKBAR M.
NOVIANA RIZA