



Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit Cloud-Service

Disusun oleh : Bihubbil Choir Aldifta



LEMBAR KESEPAKATAN

PEMBUATAN BUKU TUTORIAL “PEMBUATAN WEBSITE DINAMIS BERBASIS PHP DAN SQLITE DATABASE MENGGUNAKAN REPLIT *CLOUD-SERVICE*”

1. Nama Mahasiswa : Bihubbil Choir Aidifta (190411100121)
2. Program Studi : Teknik Informatika
3. Jenis : Menyusun Buku Tutorial
4. Topik / Judul Buku : Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit *Cloud-Service*
5. Outline Buku :
 - Pengantar Website Pengenalan Website
 - Pengantar Bahasa Web
 - Pengantar Database
 - Pengantar Replit
 - Memulai Menggunakan Replit
 - Implementasi Penggunaan Replit pada Pembuatan Website (Studi Kasus : Website Resep Makanan)

Bangkalan, Juni 2022

Menyetujui,
Dosen Pembimbing,

Pengusul,
Mahasiswa,

Dr. Noor Ifada, S.T., MISD.
NIP. 19780317 200312 2 001

Bihubbil Choir Aidifta
NIM. 190411100121

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

Cloud-Service 

LEMBAR PENGESAHAN

Telah Diperiksa dan diuji oleh dosen pembimbing pada:

Tanggal :

Dengan Nilai :

Mengetahui,

Koordinator Kerja Praktek

Dosen Pembimbing

Dr. Fika Hastarita Rachman, S.T.,
M.Eng.
NIP. 19830305 200604 2 002

Dr. Noor Ifada, S.T., MISD.
NIP. 19780317 200312 2 001

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

Cloud-Service

KATA PENGANTAR

Puji Syukur atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya. Sehingga, buku tutorial dengan judul “Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit Cloud-Service” telah selesai disusun. Buku tutorial ini disusun untuk menyelesaikan tugas Kerja Praktek, yang merupakan salah satu syarat untuk memperoleh gelar Strata Satu Jurusan Teknik Informatika Universitas Trunojoyo Madura.

Buku ini terdiri dari 6 bab utama, dimana bab pertama pada buku ini membahas tentang pengantar atau pengenalan mengenai *website*. Pada bab kedua membahas tentang pengantar atau pengenalan mengenai bahasa web. Pada bab ketiga, membahas tentang pengantar atau pengenalan mengenai *database*. Pada bab keempat membahas tentang pengantar atau pengenalan mengenai aplikasi *cloud-service* (Replit). Pada bab kelima membahas mengenai cara mendaftar dan membuat *project* pada Replit. Dan pada bab keenam buku ini membahas tentang bagaimana cara mengimplementasikan penggunaan Replit pada pembuatan *website* resep makanan, termasuk pengujiannya.

Dengan adanya buku ini penulis berharap agar buku ini dapat membantu pembaca, terutama bagi mahasiswa Teknik Informatika Universitas Trunojoyo Madura. Kritik dan saran dari pembaca sangat diperlukan guna dijadikan acuan oleh penulis untuk membuat buku atau karya yang lebih baik lagi kedepannya. Akhir kata penulis ucapkan terima kasih yang sebesar-besarnya.

Bangkalan, ... Juni 2022
Penyusun

Bihubbil Choir Aidifta

DAFTAR ISI

Lembar Kesepakatan.....	i
Lembar Pengesahan.....	ii
Kata Pengantar	iii
Daftar Isi.....	v
Daftar Gambar	ix
Daftar Tabel	xiv
Bab 1 Pengantar Website.....	1
1.1 Sejarah Website.....	1
1.2 Jenis Website	3
Bab 2 Pengantar Bahasa Web	16
2.1 HTML.....	16
2.2 CSS	21
2.3 PHP.....	32
Bab 3 Pengantar Database	38
3.1 Pengertian Database	38
3.2 Jenis Database	38
3.3 SQLite.....	41
Bab 4 Pengantar Replit.....	44
4.1 Apa itu Replit.....	44

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

Cloud-Service



4.2 Fitur Replit	46
4.3 Alasan Menggunakan Replit	49
Bab 5 Memulai Menggunakan Replit	53
5.1 Mendaftar	53
5.2 Membuat Project Baru	60
Bab 6 Implementasi Penggunaan Replit pada Pembuatan Website (Studi Kasus : Website Resep Makanan)	69
6.1 Konsep Aplikasi(Website)	69
6.1.1 Gambaran Umum	69
6.1.2 <i>Use Case</i>	70
6.2 Perancangan Database	79
6.2.1 Rancangan Model <i>Database</i>	79
6.2.2 Penerapan Model <i>Database</i> Pada <i>SQLite</i>	85
6.3 Perancangan Interface	102
6.3.1 Pembuatan Modul Koneksi <i>Database</i>	102
6.3.2 Pembuatan Interface <i>Login Admin</i>	106
6.3.3 Pembuatan <i>Index Interface Admin</i>	112
6.3.4 Pembuatan <i>Interface Dashboard Admin</i>	119
6.3.5 Pembuatan <i>Interface</i> Daftar Kategori Resep	122
6.3.6 Pembuatan <i>Interface</i> Tambah Kategori Resep ..	126
6.3.7 Pembuatan <i>Interface</i> Sunting Kategori Resep ...	130
6.3.8 Pembuatan Fungsi Hapus Kategori Resep	135

6.3.9 Pembuatan <i>Interface</i> Daftar Resep	137
6.3.10 Pembuatan <i>Interface</i> Tambah Resep	141
6.3.11 Pembuatan <i>Interface</i> Sunting Resep	151
6.3.12 Pembuatan Fungsi Hapus Resep	158
6.3.13 Pembuatan <i>Interface</i> Daftar Permintaan Resep	161
6.3.14 Pembuatan Fungsi Tandai Permintaan Resep Sebagai Duplikat	167
6.3.15 Pembuatan Fungsi <i>Logout</i>	170
6.3.16 Pembuatan <i>Interface</i> Utama (Halaman Awal).	171
6.3.17 Pembuatan <i>Interface</i> Daftar Resep Berdasarkan Kategori	183
6.3.18 Pembuatan <i>Interface</i> Konten Resep	189
6.3.19 Pembuatan <i>Interface</i> Tambah Permintaan Resep	193
6.3.20 Pembuatan <i>Interface</i> Hasil Pencarian	198
6.3.21 Pembuatan <i>Interface</i> Halaman Tidak Ditemukan	202
6.3.22 Lampiran Kode CSS Tambahan & Berkas <i>Project</i>	204
6.4 Pengujian Aplikasi	205
6.4.1 Pengujian Fungsi <i>Login</i> Admin.....	205
6.4.2 Pengujian Fungsi Kelola Kategori Resep	208
6.4.3 Pengujian Fungsi Kelola Resep.....	213

6.4.3 Pengujian Fungsi Kirim Permintaan Resep	
Pengunjung	219
6.4.4 Pengujian Fungsi Kelola Daftar Permintaan Resep	
Pengunjung	221
Daftar Pustaka	228
Lampiran	230
Lampiran 1. Modul CSS Pendukung (CSS <i>External</i>).....	230
Lampiran 2. Berkas <i>Project</i> Web Resep Makanan.....	236

DAFTAR GAMBAR

Gambar 1. 1 Contoh Website Blog	7
Gambar 1. 2 Contoh Website E-Commerce/Toko Online	8
Gambar 1. 3 Contoh Website Bisnis/Perusahaan	9
Gambar 1. 4 Contoh Website Pribadi	10
Gambar 1. 5 Contoh Website Pemerintah	11
Gambar 1. 6 Contoh Website Majalah/Portal Berita	12
Gambar 1. 7 Contoh Website Non-Profit	14
Gambar 1. 8 Contoh Website Portfolio	15
Gambar 2. 1 Logo HTML 5	16
Gambar 2. 2 Output Contoh Sintaks Dasar HTML	20
Gambar 2. 3 Logo CSS	21
Gambar 2. 4 Output Sintaks Dasar CSS	31
Gambar 2. 5 Output Sintaks Dasar PHP(Sebelum Input Nama Pada Form)	37
Gambar 2. 6 Output Sintaks Dasar PHP(Setelah Input Nama Pada Form)	37
Gambar 3. 1 Logo SQLite	41
Gambar 4. 1 Logo Replit	44
Gambar 5. 1 Halaman Utama Replit	54
Gambar 5. 2 Tombol Signup Replit	54
Gambar 5. 3 Halaman Pendaftaran Replit	55
Gambar 5. 4 Form Pendaftaran Replit	56
Gambar 5. 5 Tombol Pendaftaran Replit Melalui Layanan Populer	57
Gambar 5. 6 Form Penyiapan Profil Replit	58
Gambar 5. 7 Pemilihan Tema Replit	59

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

Gambar 5. 8 Halaman Dashboard Replit	61
Gambar 5. 9 Tombol Create Project / Repl	62
Gambar 5. 10 Form Pemilihan Bahasa Pemrograman	63
Gambar 5. 11 Form Pemberian Nama dan Pemilihan Setelah Privasi Repl.....	64
Gambar 5. 12 Halaman IDE Project/Repl Replit	65
Gambar 5. 13 Tombol Run Project/Repl	66
5. 14 Console Project/Repl Replit	67
Gambar 5. 15 Tombol Open In New Tab Mini Browser Replit	67
Gambar 5. 16 Tampilan Output Project/Repl Pada Tab Baru	68
Gambar 6. 1 Use Case	71
Gambar 6. 2 Conceptual Data Model (CDM).....	80
Gambar 6. 3 Pysical Data Model (PDM)	84
Gambar 6. 4 Shell Project/Repl Replit	87
Gambar 6. 5 Perintah Membuat Direktori Database	88
Gambar 6. 6 Hasil Perintah Membuat Direktori Database (File Manager)	89
Gambar 6. 7 Tampilan CLI Pemilihan Versi SQLite3	90
Gambar 6. 8 Tampilan CLI Setelah Memilih Versi SQLite3 ...	91
Gambar 6. 9 Hasil Perintah Membuat/Membuka Berkas Database SQLite (Shell)	92
Gambar 6. 10 Hasil Perintah Membuat/Membuka Berkas Database SQLite (File Manager)	93
Gambar 6. 11 Tampilan CLI Error “Malformed Database Disk Image”	101
Gambar 6. 12 Cara Membuat/Menambahkan Berkas Baru Menggunakan File Manager	103
Gambar 6. 13 Membuat Berkas conndB.php	103

6. 14 Tampilan Kode Pada Berkas conndB.php (Code Editor)	105
Gambar 6. 15 Tombol Refresh Mini Browser Replit.....	106
Gambar 6. 16 Mengubah Tema Replit Menjadi Mode Gelap	107
Gambar 6. 17 Membuat Berkas login.php	108
Gambar 6. 18 Membuat Direktori admin.....	112
Gambar 6. 19 Membuat Berkas admin/index.php.....	112
Gambar 6. 20 Membuat Direktori admin/content.....	120
Gambar 6. 21 Membuat Berkas admin/content/dashboard.php	120
Gambar 6. 22 Membuat Berkas admin/content/kategori.php	123
Gambar 6. 23 Membuat Berkas admin/content/tambah_kategori.php.....	126
Gambar 6. 24 Membuat Berkas admin/content/edit_kategori.php	131
Gambar 6. 25 Membuat Berkas admin/content/hapus_kategori.php	135
Gambar 6. 26 Membuat Berkas admin/content/resep.php	138
Gambar 6. 27 Membuat Berkas admin/content/tambah_resep.php	142
Gambar 6. 28 Membuat Direktori assets	150
Gambar 6. 29 Membuat Direktori image/assets	150
Gambar 6. 30 Membuat Berkas admin/content/edit_resep.php	151
Gambar 6. 31 Membuat Berkas admin/content/hapus_resep.php.....	158
Gambar 6. 32 Membuat Berkas admin/content/permintaan.php.....	161

Gambar 6. 33 Membuat Berkas	
admin/content/permintaan_duplikat.php.....	168
Gambar 6. 34 Membuat Berkas logout.php.....	170
Gambar 6. 35 Membuka Berkas index.php.....	172
Gambar 6. 36 Membuat Berkas cat_list.php.....	184
Gambar 6. 37 Membuat Berkas blog_konten.php.....	190
Gambar 6. 38 Membuat Berkas request.php.....	193
Gambar 6. 39 Membuat Berkas search.php.....	199
Gambar 6. 40 Membuat Berkas notfound.php.....	203
Gambar 6. 41 Menginputkan Kredensial Salah.....	206
Gambar 6. 42 Menginputkan Kredensial Benar.....	207
Gambar 6. 43 Menambahkan Kategori Resep.....	208
Gambar 6. 44 Hasil Penambahan Kategori Resep.....	209
Gambar 6. 45 Menyunting Kategori Resep.....	210
Gambar 6. 46 Hasil Sunting Kategori Resep.....	211
Gambar 6. 47 Hasil Hapus Kategori Resep.....	212
Gambar 6. 48 Menambahkan Resep.....	214
Gambar 6. 49 Hasil Tambah Resep.....	215
Gambar 6. 50 Menyunting Resep.....	216
Gambar 6. 51 Hasil Sunting Resep.....	217
Gambar 6. 52 Hasil Hapus Resep.....	218
Gambar 6. 53 Menambahkan Permintaan Resep.....	219
Gambar 6. 54 Hasil Penambahan Permintaan Resep.....	220
Gambar 6. 55 Menambahkan Permintaan Resep Menjadi Resep Baru.....	222
Gambar 6. 56 Hasil Penambahan Permintaan Resep Menjadi Resep Baru (Data Permintaan Resep).....	223
Gambar 6. 57 Hasil Penambahan Permintaan Resep Menjadi Resep Baru (Data Resep).....	223

Gambar 6. 58 Menandai Permintaan Resep Sebagai Duplikat	225
Gambar 6. 59 Hasil Tandai Permintaan Resep Sebagai Duplikat.....	226

DAFTAR TABEL

Tabel 2. 1 Contoh Sintaks Dasar HTML	18
Tabel 2. 2 Contoh Sintaks Dasar Inline CSS	24
Tabel 2. 3 Contoh Sintaks Dasar Internal/Embedded CSS	27
Tabel 2. 4 Contoh Sintaks Dasar External CSS(File .html)	29
Tabel 2. 5 Contoh Sintaks Dasar External CSS(File .css).....	30
Tabel 2. 6 Contoh Implementasi PHP	34
Tabel 2. 7 Contoh Implementasi PHP (Membuat Form)	35
Tabel 2. 8 Contoh Implementasi PHP (Membuat Output Teks Dinamis).....	36
Tabel 6. 1 Perintah Membuat Direktori Database.....	88
Tabel 6. 2 Perintah Memanggil/Menjalankan Aplikasi SQLite3 Berdasarkan CLI	89
Tabel 6. 3 Perintah Untuk Membuat/Membuka Berkas Database SQLite	91
Tabel 6. 4 Konsep Table ADMIN	94
Tabel 6. 5 Perintah SQL Untuk Membuat Table Admin	94
Tabel 6. 6 Konsep Table KATEGORIRESEP	95
Tabel 6. 7 Perintah SQL Untuk Membuat Table Kategori Resep	95
Tabel 6. 8 Konsep Table STATUSPERMINTAAN	96
Tabel 6. 9 Perintah SQL Untuk Membuat Table Status Permintaan	96
Tabel 6. 10 Konsep Table RESEP.....	97
Tabel 6. 11 Perintah SQL Untuk Membuat Table Resep.....	99
Tabel 6. 12 Konsep Table PERMINTAANRESEP.....	99
Tabel 6. 13 Perintah SQL Untuk Membuat Table Permintaan Resep	101

Tabel 6. 14 Perintah Untuk Menyimpan Hasil Perubahan Database Pada Berkas Baru	101
Tabel 6. 15 Kode Untuk Berkas conndB.php	104
Tabel 6. 16 Kode Untuk Berkas index.php (Uji Coba Kode Pada Berkas conndB.php)	106
Tabel 6. 17 Kode Untuk Berkas login.php	110
Tabel 6. 18 Kode Untuk Berkas admin/index.php	118
Tabel 6. 19 Kode Untuk Berkas admin/content/dashboard.php	122
Tabel 6. 20 Kode Untuk Berkas admin/content/kategori.php	125
Tabel 6. 21 Kode Untuk Berkas admin/content/tambah_kategori.php	129
Tabel 6. 22 Kode Untuk Berkas admin/content/edit_kategori.php	134
Tabel 6. 23 Kode Untuk Berkas admin/content/hapus_kategori.php	136
Tabel 6. 24 Kode Untuk Berkas admin/content/resep.php	140
Tabel 6. 25 Kode Untuk Berkas admin/content/tambah_resep.php	148
Tabel 6. 26 Kode Untuk Berkas admin/content/edit_resep.php	157
Tabel 6. 27 Kode Untuk Berkas admin/content/hapus_resep.php	160
Tabel 6. 28 Kode Untuk Berkas admin/content/permintaan.php	165
Tabel 6. 29 Kode Untuk Berkas admin/content/permintaan_duplikat.php	169
Tabel 6. 30 Kode Untuk Berkas logout.php	171

Tabel 6. 31 Kode Untuk Berkas index.php (Kode Halaman Utama)	182
Tabel 6. 32 Kode Untuk Berkas cat_list.php.....	188
Tabel 6. 33 Kode Untuk Berkas blog_konten.php	192
Tabel 6. 34 Kode Untuk Berkas request.php	197
Tabel 6. 35 Kode Untuk Berkas search.php.....	202
Tabel 6. 36 Kode Untuk Berkas notfound.php	204

BAB 1

PENGANTAR WEBSITE

1.1 Sejarah Website

Website atau situs merupakan suatu halaman yang sering kali kita kunjungi saat berselancar di internet atau dunia maya, entah itu ketika mencari informasi, mencari hiburan ataupun belanja online Internet.

Menurut bahasa, website (atau disingkat web) adalah suatu system yang digunakan untuk mengakses, memanipulasi, serta mengunduh dokumen hyperlink yang terdapat pada komputer yang terhubung melalui internet atau jaringan.

Berdasarkan laman History of Webiste, salah satu orang yang cukup berjasa menemukan website adalah Sir Tim Berners-Lee. Dia adalah Ilmuwan komputer asal Inggris dan merupakan insinyur software (perangkat lunak) pada CERN –laboratorium fisika (partikel besar) yang berlokasi di dekat Jenewa, Swiss. Penemuan website tersebut awalnya dipelopori oleh permasalahan yang dihadapi Berners, yang mana pada saat itu dia merasa kesulitan ketika ingin berbagi informasi satu sama lain.

Di Tahun 1990, tepatnya pada bulan oktober, Berners membuat tulisan mengenai tiga teknologi dasar web, yang mana ketiganya masih digunakan sampai saat ini sebagai dasar pembuatan web.

Ketiganya antara lain :

- HTML (HyperText Markup Language), merupakan bahasa dengan markup format, yang umum digunakan dalam pembuatan web.
- URI(Uniform Resource Identifier), merupakan “alamat” unik yang berfungsi untuk mengidentifikasi tiap-tiap resource (sumber daya) yang ada di web. Kini lebih awam disebut dengan URL.
- HTTP(Hypertext Transfer Protocol), merupakan protocol transfer yang memungkinkan kita untuk mengambil kembali sumber daya yang terhubung dari seluruh web.

Selain itu, Berners juga membuat halaman web editor/browser pertama, yaitu (“WorldWideWeb.app”) dan juga server web pertama, yaitu (“httpd”) web server.

Pada akhir 1990, halaman web pertama akhirnya dapat tersedia di internet secara terbuka. Pada 1991, orang-orang eksternal (diluar CERN, dan termasuk didalamnya orang awam), juga

turut diundang untuk bergabung dalam komunitas web baru ini. Dari situlah, awal mulanya lahir website sebagai media dan juga alat komunikasi otomatis yang dapat digunakan untuk berbagi informasi antar ilmuwan yang ada di berbagai belahan dunia.

Kemudian barulah pada 6 Agustus 1991, tim Berners membuat website terbuka untuk publik. Dan pada 1994, Berners kembali mendirikan sebuah group bernama “World Wide Web Consortium”, group ini bertujuan untuk melakukan pengembangan protokol dan guideline guna memastikan pertumbuhan jangka panjang dari website.

1.2 Jenis Website

Sebelum memulai membuat sebuah website, tentu perlu diketahui jenis website seperti apa yang dibutuhkan. Pada saat ini, terdapat berbagai macam jenis website yang ada di internet, jenis website tersebut dapat dibagi ke dalam beberapa kategori, salah satunya dapat dibagi berdasarkan sifat dan fungsinya.

Terdapat berbagai hal yang sebaiknya perlu dipertimbangkan guna dapat menemukan produk yang paling tepat untuk kita.

Oleh sebab itu, terdapat juga bermacam-macam jenis website, yang mana setiap macamnya membawa fungsi yang berbeda.

- **Jenis Website Berdasarkan Sifat**

- Website Statis

Secara sederhana, website statis dapat diartikan sebagai sebuah website yang isi atau kontennya konstan (tidak berubah). Setiap halaman yang ada didalam website statis umumnya dibuat menggunakan kode HTML, oleh karena itu, tiap informasi yang ditampilkan kepada setiap pengunjungnya, semuanya akan sama. Hanya pengembang atau webmaster sendiri lah yang dapat megubah tampilan dan konten yang ada pada website statis.

Secara umum, website statis biasanya dibuat menggunakan bahasa HTML dan CSS, oleh pengembang atau webmaster yang memahami bahasa pemrograman dan coding. Alih-alih membuat kodingan dari awal, saat ini terdapat beberapa layanan penyedia tool yang dapat digunakan untuk melakukan generator website statis, beberapa diantaranya seperti Jekyll, Hexo, atau Hugo.

Karena website jenis ini tidak membutuhkan adanya perubahan konten secara berkala, maka penggunaan database tidaklah diperlukan dalam implementasinya. Website statis biasanya digunakan untuk membuat website perusahaan yang hanya perlu memberikan informasi-informasi statis seperti alamat, kontak, dan sejarah perusahaan.

- Website Dinamis

Berbeda dengan website statis yang kontennya selalu konstan atau tidak berubah-ubah, website dinamis merupakan website yang kontennya selalu berubah atau di-update secara berkala. Kebanyakan website yang ada saat ini bersifat dinamis, hal ini disebabkan karena website dinamis lebih mudah dikelola, jika dibandingkan dengan website statis.

Website dinamis dapat menampilkan konten sesuai apa yang ada pada database dan biasanya hanya bisa diakses oleh pengembang atau webmaster. Namun hal ini juga tidak menutup kemungkinan bahwa pengguna juga dapat melakukan pembaruan konten, jika pengembang

menyediakan interface tersendiri untuk pengguna yang berfungsi untuk melakukan pembaruan konten tertentu.

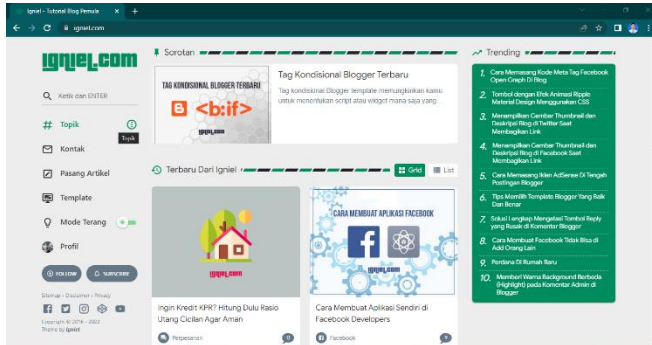
- Jenis Website Berdasarkan Fungsi

1. Blog

Blog merupakan jenis website yang didalamnya berisi postingan mengenai konten-konten tertentu, misalnya artikel, gambar, dan juga video.

Website jenis ini biasanya berfungsi untuk memposting tulisan yang sifatnya personal, artikel atau pembahasan mengenai suatu hal, hingga media promosi konten untuk kepentingan bisnis.

Contoh Website berjenis Blog : mastimon.com dan igniel.com.



Gambar 1. 1 Contoh Website Blog

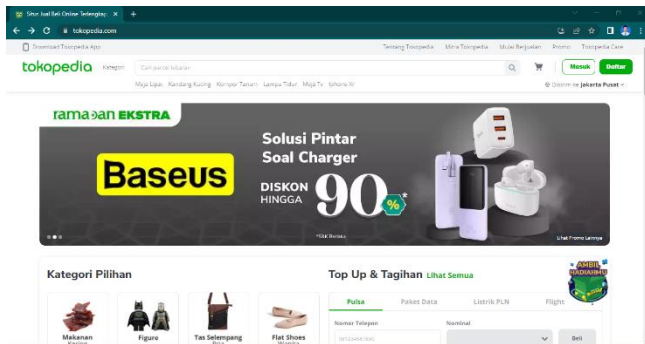
2. E-commerce/Toko Online

E-commerce atau toko online merupakan jenis website yang berfungsi sebagai tempat jual beli atau bisa disebut toko virtual. Jadi, pada toko tersebut pengunjung akan melihat beragam produk atau layanan yang tersedia di dalamnya.

Apabila kita ingin membangun suatu bisnis, toko online sepertinya bisa menjadi ide yang tepat untuk mengembangkan bisnis kita. Karena jika dikelola

dengan tepat, kita akan bisa meraih banyak pembeli baru di website kita.

Contoh Website E-commerce : tokopedia.com dan eigeradventure.com



Gambar 1. 2 Contoh Website E-Commerce/Toko Online

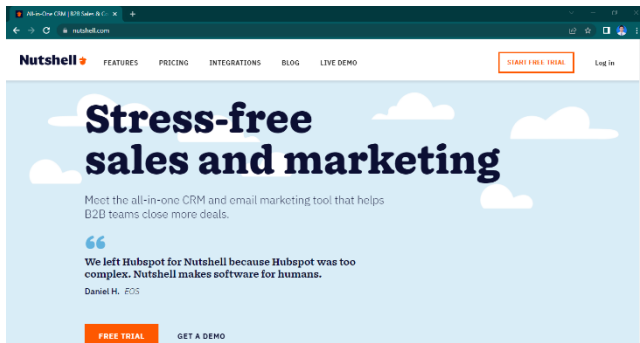
3. Website Bisnis/Perusahaan

Website bisnis atau perusahaan merupakan jenis website yang berfungsi untuk memperkenalkan bisnis atau perusahaan kita kepada public secara digital.

Dengan membuat website sejenis ini, kita dapat menjelaskan tentang sejarah, produk atau jasa yang

ditawarkan, serta nilai-nilai yang dipegang pada bisnis atau perusahaan.

Contoh Website Bisnis/Perusahaan : studioneat.com dan nutshell.com



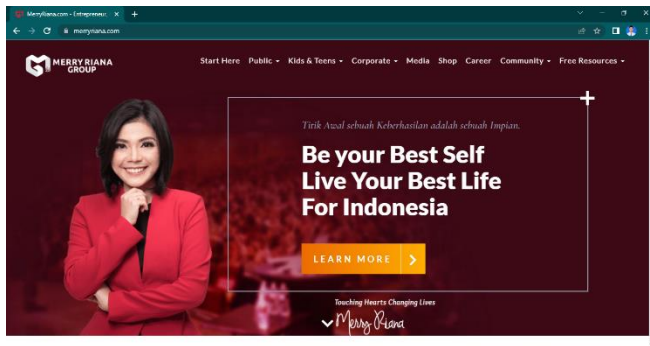
Gambar 1. 3 Contoh Website Bisnis/Perusahaan

4. Website Pribadi

Website pribadi merupakan jenis website yang berfungsi untuk berbagi berbagai hal tentang pribadi seseorang atau personal branding. Jenis website ini biasanya digunakan digunakan oleh public figure, seperti selebritis atau politikus.

Meski memang sering diggunakan oleh public figure, hal ini bukan berarti kita harus menjadi public figure juga untuk membuatnya. Kita juga bisa membuat website jenis ini dengan mengisi aktivitas-aktivitas kita, yang menurut kita menarik untuk dipublikasikan ke khalayak.

Contoh Website Pribadi : merryriana.com



Gambar 1. 4 Contoh Website Pribadi

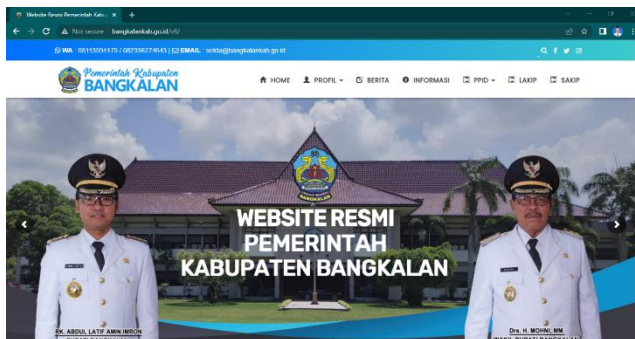
5. Website Pemerintah

Website pemerintah merupakan jenis website yang berfungsi untuk menampilkan berbagai informasi tentang suatu instansi pemerintahan, contohnya seperti informasi mengenai layanan, berita instansi, serta

program digital yang disediakan suatu instansi pemerintah.

Salah satu yang membedakan website jenis ini dengan website jenis lainnya biasanya adalah pada ekstensi domainnya, yang mana biasanya ekstensi domain yang digunakan memang khusus untuk instansi pemerintah, contohnya adalah seperti “gov.id” atau “.gov”

Contoh Website Pemerintah : bangkalankab.go.id



Gambar 1. 5 Contoh Website Pemerintah

6. Website Majalah/ Portal Berita

Website majalah atau portal berita merupakan jenis website yang berfungsi untuk menyampaikan

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

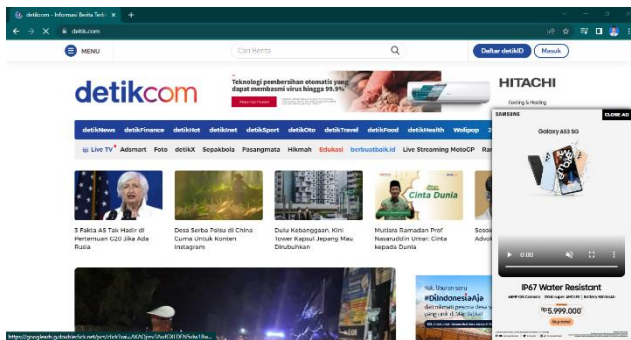
Cloud-Service

11

informasi mengenai trend atau berita yang saat ini sedang terjadi ataupun sedang hangat diperbincangkan pada masyarakat masyarakat.

Jenis web ini sebenarnya memiliki konsep yang hampir sama dengan blog, yaitu berisi postingan-postingan mengenai artikel. Perbedaanya adalah ada pada jenis artikel yang diposting, yaitu cenderung melakukan postingan artikel mengenai berita atau trend yang sedang terjadi.

Contoh Website Majalah atau Berita :
majalah.tempo.co dan detik.com



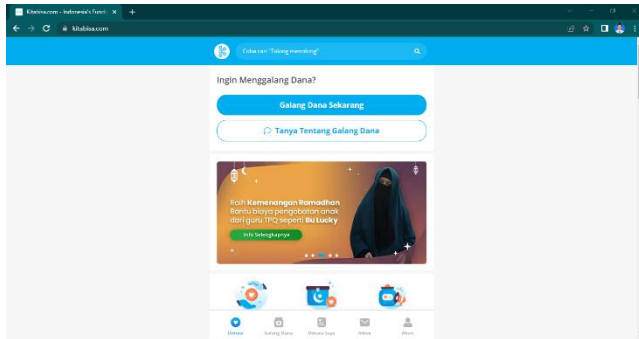
Gambar 1. 6 Contoh Website Majalah/Portal Berita

7. Website Non-profit

Website Non-profit merupakan jenis website yang berfungsi sebagai sarana suatu organisasi untuk melakukan kegiatan non-profitnya secara digital.

Salah satu contoh kegiatan non-profit yang sering diterapkan pada sebuah website non-profit adalah seperti penggalangan dana atau donasi untuk bencana alam ataupun suatu permasalahan spesifik lainnya.

Contoh Website Non-Profit : kitabisa.com



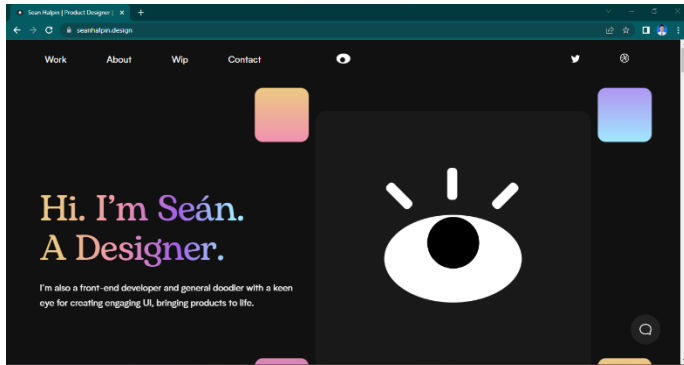
Gambar 1. 7 Contoh Website Non-Profit

8. Website Portfolio

Website portfolio merupakan jenis website yang berfungsi untuk mempromosikan hasil karya dan pengalaman kerja seseorang.

Dengan menggunakan jenis ini, pemilik website dapat berpeluang mendapatkan pekerjaan ataupun proyek melalui seseorang atau perusahaan yang melihat preview hasil karya atau pengalaman kerja yang ditampilkan pada website portfolio.

Contoh Website Portfolio : seanhalpin.design



Gambar 1. 8 Contoh Website Portfolio

BAB 2

PENGANTAR BAHASA WEB

2.1 HTML



Gambar 2. 1 Logo HTML 5

Hypertext Markup Language atau yang lebih *familiar* disingkat HTML, merupakan sebuah bahasa markah yang sangat populer digunakan untuk membuat sebuah halaman web saat ini. Untuk melakukan decode atau menampilkan bahasa markup ini kedalam bentuk yang dapat dicerna dengan mudah oleh khalayak, umumnya digunakan sebuah aplikasi peramban atau *browser*.

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

Dalam penulisannya, HTML ditulis dalam format ASCII (American Standard Code for Information Interchange), yaitu format standar bahasa encoding amerika yang memang dirancang untuk melakukan pertukaran informasi antara computer dan internet. Kelebihan menggunakan format ini adalah karena sifatnya yang universal, jadi hampir semua komputer modern akan dapat mengenali dan membaca format ini.

Pada HTML, terdapat yang namanya *hypertext link* atau sebuah tautan *hypertext*, yang mana fungsinya adalah menautkan antara teks dengan suatu dokumen. Dengan demikian, kita dapat memanfaatkan fitur ini untuk dapat berpindah ke dokumen lain, hanya dengan mengklik tautan yang disematkan pada suatu teks.

Contoh penulisan sintaks dasar HTML adalah sebagai berikut.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Ini adalah judul
Halaman</title>
    </head>
    <body>
        <h1>Ini adalah body</h1>
        <p>
            body digunakan untuk
mendefinisikan apasaja data yang akan
ditampilkan pada halaman html, semua tag
yang ditambahkan pada body akan dapat
ditampilkan pada peramban.
        </p>
    </body>
</html>
```

Tabel 2. 1 Contoh Sintaks Dasar HTML

Keterangan tag:

- <!DOCTYPE html> :

Tag yang digunakan untuk memberikan informasi kepada peramban mengenai tipe (termasuk versi) dokumen yang digunakan, pada kasus ini yang digunakan adalah dokumen tipe html 5.

- `<html>` :

Tag awal yang digunakan untuk mendefinisikan sebuah dokumen dengan tipe html.

- `<head>` :

Tag yang digunakan untuk mendefinisikan berbagai informasi/metadata yang akan diterapkan pada dokumen html. Tag-tag yang berada pada `<head>` tag umumnya tidak akan ditampilkan pada halaman saat dibuka.

- `<title>` :

Tag yang digunakan untuk mendefinisikan judul dari sebuah dokumen html, judul dari dokumen ini biasanya akan muncul pada fitur tab di peramban.

- `<body>` :

Tag yang digunakan untuk mendefinisikan apa saja data yang akan ditampilkan pada halaman html, semua tag yang

ditambahkan pada body akan dapat ditampilkan pada peramban.

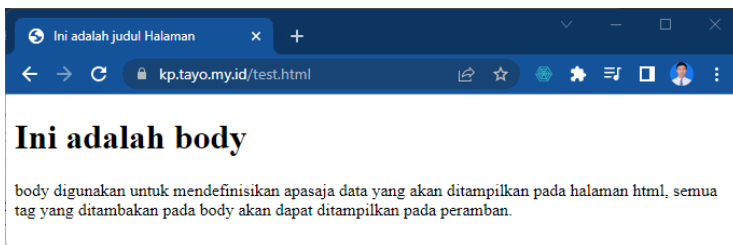
- `<h1>` :

Merupakan tag *heading*, biasanya digunakan untuk menampilkan suatu judul dari halaman atau suatu informasi yang akan ditampilkan.

- `<p>` :

Merupakan tag *paragraph*, biasanya digunakan untuk menyampaikan isi atau bahasan dari suatu topik atau judul.

Dari contoh sintaks dasar diatas, berikut adalah *output* yang akan ditampilkan jika kita membukanya pada peramban.



Gambar 2. 2 Output Contoh Sintaks Dasar HTML

2.2 CSS



Gambar 2. 3 Logo CSS

Cascading Style Sheet atau yang lebih *familiar* disingkat CSS merupakan sebuah aturan yang digunakan untuk mengendalikan tampilan komponen-komponen yang ada pada sebuah web, sehingga dapat menampilkan tampilan yang lebih terstruktur, menarik, dan juga seragam.

CSS sendiri dibuat oleh perkumpulan/komunitas web dunia, yaitu *World Wide Web Consortium*(W3C). Cara kerja CSS

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

adalah dengan cara menerapkan aturan *style/gaya* pada komponen-komponen yang ada di dokumen(umumnya HTML), dengan memanfaatkan fitur yang dinamakan *css selector*.

CSS selector merupakan salah satu fitur pada CSS, yang mana fungsinya adalah untuk memberikan batasan pada komponen apa saja yang akan diterapkan aturan *style/gaya* tertentu, jadi misal kita menerapkan suatu aturan *style/gaya* dengan menggunakan selector pada *element p*, maka seluruh komponen pada *element p* yang ada pada dokumen akan terpengaruh oleh aturan *style/gaya* yang diterapkan tersebut. Sementara untuk jenisnya, CSS memiliki beberapa jenis selector, beberapa diantaranya adalah *element selector*, *id selector*, *class selector*, *attribute selector*, dan juga *global selector*.

Jenis-Jenis Cascading Style Sheet (CSS)

Pada penggunaannya, terdapat 3 jenis css yang dapat kita terapkan untuk memberikan *style/gaya* pada komponen yang ada pada dokumen HTML, ketiganya adalah sebagai berikut.

- *Inline CSS*

Inline CSS merupakan jenis css yang penggunaannya dengan cara meletakkan aturan *style/gaya* langsung pada *tag element* yang dituju.

Sebagai contoh, misal kita ingin menerapkan css background berwarna kuning pada *element* body dan tulisan berwarna biru pada *element* h1 di sintaks dasar html sebelumnya, maka aturan sintaksnya akan seperti berikut.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Ini adalah judul
        Halaman</title>
```



```

        </head>
        <body style="background-color :
yellow;">

            <h1 style="color:
blue;">Ini adalah body</h1>

            <p>
                body digunakan untuk
mendefinisikan apasaja data yang akan
ditampilkan pada halaman html, semua tag
yang ditambahkan pada body akan dapat
ditampilkan pada peramban.

            </p>

        </body>
</html>

```

Tabel 2. 2 Contoh Sintaks Dasar Inline CSS

Keterangan :

Untuk menerapkan inline CSS, kita harus menambahkan *attribute* “*style*” didalam tag element yang ingin kita ubah tampilannya, kemudian untuk value *atributenya*, kita dapat mengisi dengan aturan style/gaya CSS yang kita inginkan.

Misal pada penerapan sintaks di atas :

- `<body>`, **menjadi** `<body style="background-color: yellow;">`
- `<h1>`, **menjadi** `<h1 style="color: blue;">`

- *Internal/Embedded CSS*

Berbeda dengan *Inline CSS* yang menambahkan *attribute style* langsung di pada tag element yang dituju. Pada *internal/embedded CSS*, aturan css akan diletakkan ditempat yang terpisah, yaitu dengan membuat *tag element style* tersendiri (umumnya diletakkan di dalam *tag element head*), kemudian aturan-aturam style/gaya akan diletakkan di dalam *element style* tersebut.

Untuk implementasinya, masih sama seperti sebelumnya yaitu memberikan aturan style/gaya berupa background kuning pada *element body*, dan tulisan berwarna biru pada *element h1*. Maka dengan menerapkan *internal/embedded CSS*, sintaksnya akan seperti berikut.

```
<!DOCTYPE html>

<html>

  <head>

    <title>Ini adalah judul Halaman</title>

    <style>

      body {

        background-color: yellow;

      }

      h1 {

        color : blue;

      }

    </style>

  </head>

  <body>

    <h1>Ini adalah body</h1>

    <p>

      body digunakan untuk mendefinisikan apasaja
      data yang akan ditampilkan pada halaman html,
      semua tag yang ditambahkan pada body akan dapat
      ditampilkan pada peramban.

    </p>
```

```
</body>  
</html>
```

Tabel 2. 3 Contoh Sintaks Dasar Internal/Embedded CSS

Keterangan :

Untuk menerapkan *internal/embedded* CSS, kita harus menambahkan *element style* pada dokumen HTML, kemudian barulah pada elemen style tersebut kita dapat menempatkan aturan style/gaya pada element lain, dengan memanfaatkan fitur selector pada CSS.

Misal pada penerapan sintaks di atas, untuk menerapkan aturan pada *element body* dan *element h1*, maka perlu menggunakan *element selector* pada kedua *element* tersebut.

- *Element selector* untuk body :

```
body {  
    background-color: yellow;  
}
```

- *Element selector* untuk h1 :

```
h1 {  
    color : blue;  
}
```

- External CSS

Hampir sama seperti *internal/embedded* CSS yang menempatkan aturan style pada tempat terpisah, namun pada *external* CSS, penempatan *tag stylenya* tidak langsung ditempatkan di dalam dokumen yang sama, melainkan ditempatkan pada dokumen terpisah. Kemudian barulah setelah itu, dokumen terpisah tadi yang berisi aturan CSS tersebut, akan dipanggil pada dokumen yang membutuhkan aturan CSS tersebut.

Untuk implementasinya, tetap sama seperti kedua contoh sebelumnya, yaitu memberikan aturan style/gaya berupa background kuning pada *element body*, dan tulisan berwarna biru pada *element h1*. Maka dengan menerapkan *external* CSS, sintaksnya akan seperti berikut.

```

<!DOCTYPE html>
<html>
    <head>
        <title>Ini adalah judul
        Halaman</title>
        <link rel="stylesheet"
        href="style.css">
    </head>
    <body>
        <h1>Ini adalah body</h1>
        <p>
            body digunakan untuk
            mendefinisikan apasaja data yang
            akan ditampilkan pada halaman html,
            semua tag yang ditambahkan pada body
            akan dapat ditampilkan pada
            peramban.
        </p>
    </body>
</html>

```

Tabel 2. 4 Contoh Sintaks Dasar External CSS(File .html)

style.css
<pre>body { background-color: yellow; } h1 { color : blue; }</pre>

Tabel 2. 5 Contoh Sintaks Dasar External CSS(File .css)

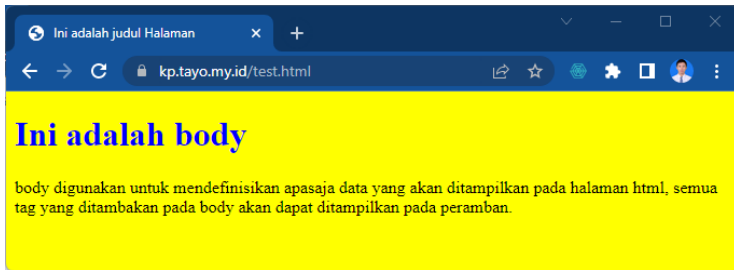
Keterangan :

Untuk menerapkan *external* CSS, kita harus menambahkan memanggil *file* dokumen yang berisi aturan CSS ke dalam dokumen saat ini atau yang ingin kita terapkan aturan CSSnya.

Misal pada penerapan sintaks di atas, untuk menerapkan aturan CSS yang ada pada *file* style.css pada dokumen saat ini , maka kita perlu memanggil file tersebut dengan cara menambahkan tag link pada bagian *header* dengan *attribute* relasi *stylesheet*, dan referensi ke file style.css seperti berikut.

- `<link rel="stylesheet" href="style.css">`

Dari ketiga jenis CSS diatas, *ouput* yang dihasilkan semuanya akan sama, yaitu seperti pada gambar berikut.



Gambar 2. 4 Output Sintaks Dasar CSS

2.3 PHP

PHP: Hypertext Preprocessor atau yang lebih familiar disebut PHP, merupakan bahasa pemrograman yang memang dirancang secara khusus untuk pembuatan dan pengembangan web. PHP sendiri dapat digunakan bersamaan dengan HTML, dan pada umumnya digunakan untuk membuat situs web yang dinamis.

PHP sendiri diciptakan pada tahun 1994 oleh Rasmus Lerdorf, hingga saat ini banyak digunakan oleh sebagian besar website yang ada di dunia. Bahkan, beberapa CMS(*Content Management System*) populer yang ada di dunia seperti Joomla!, Mambo, Postnuke, dan Xaraya, semuanya dibangun menggunakan bahasa pemrograman PHP.

Dengan menggunakan PHP, kita dapat menampilkan data pada database ke dalam website yang kita miliki. Hal ini dikarenakan PHP memiliki fungsi yang dapat mengubah serta memanggil data yang ada dalam database, kemudian menampilkannya ke dalam sebuah web.

Sebagai bahan pembelajaran, disini kita akan coba untuk mengimplementasikan contoh sederhana penggunaan PHP yang dikombinasikan dengan halaman HTML, untuk membuat sebuah halaman sederhana yang bersifat dinamis.

Untuk contoh implementasinya, disini kita akan mencoba membuat sebuah halaman yang dapat menampilkan nama sesuai yang kita *inputkan* pada *form input* nama. Berikut adalah sintaksnya.

tampilkan_nama.php

```
<html>
  <head>
    <title>Tampilkan Nama</title>
  </head>

  <body>

    <form method="post" action="">
      <label>Nama</label>
      <input type="text" name="nama"
value="">
```

```

        <input type="submit"
name="tombolsubmit">
    </form>

    <h3>Selamat datang di Website kami,
<?php echo $_POST["nama"]?></h3>
</body>

</html>

```

Tabel 2. 6 Contoh Implementasi PHP

Keterangan :

Untuk dapat menampilkan nama sesuai dengan yang diinputkan pada *form input* nama, maka yang perlu dilakukan adalah sebagai berikut.

- Membuat sebuah form dengan input nama dan tombol submit.

```

<form method="post" action="">
    <label>Nama</label>
    <input type="text" name="nama"
value="">

```

```
<input type="submit"
name="tombolsubmit">
</form>
```

Tabel 2. 7 Contoh Implementasi PHP (Membuat Form)

Catatan :

- Pada form tersebut terdapat *attribute* `method="post"` `action=""`, kedua *attribute* tersebut merupakan metode dan aksi yang akan dilakukan saat form *disubmit*. Jadi pada halaman ini form yang kita gunakan adalah form dengan metode *post*, dan untuk *action* dibiarkan kosong karena kita tidak ingin melakukan tindakan atau berpindah ke halaman lain setelah *form* ini dikirim.
- Pada element input, karena data yang akan kita inputkan berupa nama, maka tipe yang dipilih adalah *text*.
- Membuat suatu teks *element* yang nantinya akan digunakan untuk menampilkan nama yang telah dikirimkan sebelumnya pada *form input* nama.

```
<h3>Selamat datang di Website kami, <?php  
echo $_POST["nama"]?></h3>
```

Tabel 2. 8 Contoh Implementasi PHP (Membuat Output Teks Dinamis)

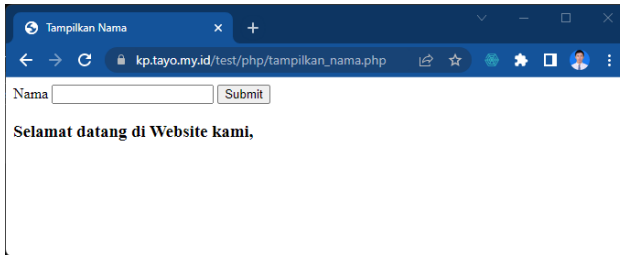
Catatan :

- Pada sintaks diatas, dibuat sebuah announcement, yang pada akhirnya terdapat sintaks PHP yaitu `echo $_POST["nama"]?>`, sintaks tersebut berfungsi untuk memanggil dan menampilkan value dari variable yang telah dikirimkan melalui form sebelumnya.

Agar *value* dari variable hasil inputan dapat tertulis dengan benar, pastikan variable post yang dipanggil (`$_POST["nama"]`) sama dengan value yang ada pada *attribute name* di *element input*, pada *form* (`<input type="text" name="nama" value="">`).

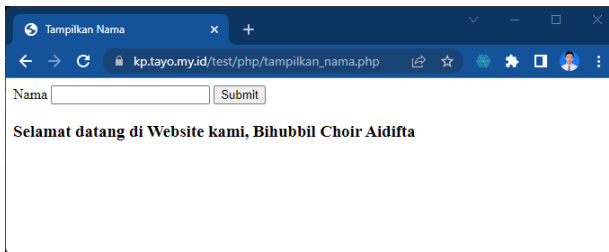
Dari sintaks/kode program untuk menampilkan nama sesuai yang *diinputkan* pada input form, berikut adalah tampilan/outputnya.

- Tampilan sebelum menginputkan nama pada *form*.



Gambar 2. 5 Output Sintaks Dasar PHP (Sebelum Input Nama Pada Form)

- Tampilan setelah menginputkan nama pada form.



Gambar 2. 6 Output Sintaks Dasar PHP (Setelah Input Nama Pada Form)

Sementara Untuk contoh implementasi PHP yang lebih spesifik, yaitu dalam pembuatan *website* resep makanan menggunakan *platform* Replit, bagian ini nantinya akan dijelaskan pada section Perancangan Interface.

BAB 3

PENGANTAR DATABASE

3.1 Pengertian Database

Basis data atau yang biasa kita sebut dengan database, merupakan sekumpulan data yang dapat saling berhubungan antara satu dengan lainnya berkat adanya pengelolaan dengan teknik dan ketentuan tertentu. Pengelolaan ini dimaksudkan agar pengguna dapat mendapat kemudahan untuk mencari, mengubah, menyimpan, dan menghapus suatu informasi yang terdapat pada kumpulan data.

Selain pengertian diatas, database juga berarti suatu sistem yang dapat mengumpulkan berkas, tabel, ataupun suatu arsip yang terhubung dan tersimpan di suatu media, khususnya media elektronik.

3.2 Jenis Database

Dalam penerapannya, saat ini terdapat beberapa jenis *database* yang aktif digunakan atau beroperasi hingga saat ini. Tiap jenis database tersebut tentunya memiliki fungsi yang berbeda-beda dan memang dibuat sesuai dengan kebutuhan tertentu.

Berikut adalah contoh jenis database yang aktif dan beroperasi hingga saat ini :

- ***Operational Database***

Operational database atau yang lebih dikenal dengan nama OLTP (*On Line Transaction Processing*) database, merupakan jenis *database* yang berguna untuk mengelola suatu data secara dinamis dan *real-time* atau langsung.

Contoh aplikasi yang menerapkan konsep operational database : JSON dan XML.

- ***Warehouse Database***

Warehouse database merupakan jenis *database* yang berfungsi sebagai wadah atau repositori berbagai data yang nantinya dapat digunakan untuk melakukan pelaporan dan analisis data.

Contoh aplikasi yang menerapkan konsep *warehouse database* : Microsoft SQL Server.

- ***Distributed Database***

Distributed database merupakan jenis database yang berfungsi untuk dapat menyimpan data di beberapa tempat yang berbeda, database jenis ini biasanya berada di beberapa atau banyak sistem yang berbeda dan saling terhubung.

Contoh aplikasi yang menerapkan konsep *distributed database* : Microsoft Office Access

- ***Relational Database***

Relational database merupakan jenis database yang berfungsi untuk mengorganisir beberapa data yang saling berhubungan antara satu dengan lainnya. Database jenis ini umumnya menggunakan Structured Query Language (SQL) sebagai bahasa pemrograman untuk pengelolaan datanya.

Contoh aplikasi yang menerapkan konsep *relational database* : MySQL, PostgreSQL, dan MariaDB

- ***End-User Database***

End-user database merupakan jenis database yang tempat penyimpanan datanya langsung pada penyimpanan lokal, jadi tidak seperti mesin *database client server*. Jenis database ini paling sering digunakan pada peramban web dan sistem operasi, namun tidak jarang juga digunakan sebagai *database website*.

3.3 SQLite



Gambar 3. 1 Logo SQLite

SQLite merupakan DBMS atau *Database Management System* yang dibangun menggunakan bahasa pemrograman C. DBMS ini sendiri sangat populer digunakan sebagai aplikasi atau perangkat lunak untuk menyimpan data pada *database* secara lokal.

Sesuai dengan namanya, SQLite sendiri menggunakan sebagian besar bahasa SQL, namun ada beberapa fitur yang dihilangkan.

Berikut adalah beberapa fitur SQL yang dihilangkan pada SQLITE :

- **RIGHT dan FULL OUTER JOIN**

Untuk **LEFT OUTER JOIN** tetap diimplementasikan, sementara untuk **RIGHT OUTER JOIN** dan **FULL OUTER JOIN** dihilangkan.

- **Operasi ALTER TABLE**

Hanya perintah **RENAME TABLE**, **ADD COLUMN**, **RENAME COLUMN**, dan **DROP COLUMN** dari perintah **ALTER TABLE** yang didukung. Jenis operasi ALTER TABLE lainnya seperti ALTER COLUMN, ADD CONSTRAINT dihilangkan.

- **Write Access pada VIEWS**

Hak akses pada **VIEWS** hanya bersifat *read-only*. Jadi kita tidak dapat menjalankan *statement* **INSERT**, **DELETE**, dan **UPDATE** pada **VIEWS**.

- **GRANT dan REVOKE Permission**

Karena SQLite menyimpan berkas *database* pada penyimpanan lokal, jadi perintah **GRANT** dan **REVOKE** *permission* yang biasanya ada pada *client/server* DBMS tidak akan berfungsi.

Untuk contoh implementasi dan penggunaan aplikasi SQLite, bagian ini nantinya akan dijelaskan pada section 6.2.2 Penerapan Model *Database* Pada *SQLite*. Bagian ini akan langsung menjelaskan mengenai contoh implementasi dan penggunaan SQLite pada pembuatan *database website* resep makanan menggunakan *platform* Replit, sehingga dapat memberikan pemahaman yang lebih spesifik kepada pembaca tentang penggunaan SQLite pada *platform* Replit.

BAB 4

PENGANTAR REPLIT

4.1 Apa itu Replit



Gambar 4. 1 Logo Replit

Replit merupakan cloud-based service atau layanan berbasis awan yang fungsi utamanya adalah sebagai platform untuk menulis kode dan menghosting aplikasi dari berbagai bahasa pemrograman.

Sesuai dengan jenisnya yaitu aplikasi berbasis awan atau *cloud-based service*, jadi untuk mengakses seluruh layanan Replit, yang kita butuhkan hanyalah sebuah peramban dan koneksi internet. Jadi selama sistem operasi yang kita gunakan

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

mendukung koneksi internet dan aplikasi peramban, kita dapat menggunakannya untuk mengakses layanan Replit.

Untuk dukungan bahasa pemrograman, Replit sendiri telah mendukung lebih dari 50 bahasa pemrograman. Sebagai contoh, berikut adalah beberapa bahasa pemrograman populer yang didukung oleh Replit :

- Python
- C
- PHP
- JavaScript
- SQL (Menggunakan SQLite)
- Swift
- Kotlin

Selain bahasa pemrograman, Replit juga mendukung bahasa web seperti HTML dan CSS, yang mana hal ini akan menjadikan Replit sebagai *platform* yang cocok untuk melakukan pengembangan aplikasi berbasis web.

4.2 Fitur Replit

Setelah Mengetahui apa itu Replit, selanjutnya akan dijelaskan mengenai fitur-fitur yang ada pada Replit. Untuk fiturnya sendiri, Replit memiliki beberapa fitur menarik seperti :

- ***In-Browser IDE***

Sebagai platform untuk menulis kode, sebenarnya Replit juga dapat berfungsi sebagai *Integrated Development Environment* (IDE). Perlu diketahui, bahwa IDE sendiri memiliki fitur yang lebih daripada *code editor* pada umumnya, hal ini karena selain berfungsi sebagai code editor, IDE juga memiliki fitur *compiler*, *code debugger*, serta *syntax highlighter*.

Dengan adanya fungsi IDE ini, tentunya Replit akan lebih mendukung untuk membuat dan mengembangkan suatu perangkat lunak, terutama bagi mereka yang sering menggunakan platform sistem operasi yang berbeda-beda. karena mereka dapat langsung mengaksesnya melalui *browser* tanpa harus menginstall aplikasi tambahan .

- ***Host & deploy***

Untuk fungsinya yang dapat digunakan untuk meng*hosting* aplikasi, sebenarnya fungsi ini juga merupakan pelengkap dari fungsi sebelumnya, yaitu untuk menulis kode. Karena dengan menggunakan Replit, selain dapat digunakan untuk membuat dan mengembangkan aplikasi, kita juga dapat menggunakan Replit sekaligus sebagai sarana untuk meng*hosting* aplikasi yang telah kita buat atau kembangkan tadi.

Sebagai contoh misal kita akan membuat atau mengembangkan suatu *website* menggunakan Replit, disaat yang sama saat mengembangkan aplikasi tersebut, sebenarnya kita juga langsung mende*ploy* aplikasi yang kita buat atau kembangkan tadi secara *live* pada platform Replit. Jadi sambil mengembagkan suatu aplikasi, kita juga dapat langsung mengaksesnya pada *platform hosting* yang disediakan Replit.

- ***Real-time collaboration***

Masih berkaitan dengan fungsinya sebagai IDE, Replit dapat digunakan untuk mengembangkan suatu perangkat lunak atau aplikasi bersama orang lain dan secara *live* atau *real-time*. Oleh karena itu, Replit sendiri akan sangat

nyaman digunakan, terutama bagi mereka yang sering membuat atau mengembangkan suatu perangkat lunak bersama dengan orang lain atau tim.

- ***Learning and Teaching Platform***

Selain platform untuk membuat atau mengembangkan suatu aplikasi dengan bahasa pemrograman tertentu, Replit juga dapat menjadi platform untuk sarana belajar dan mengajar.

Replit dapat dijadikan sebagai sarana belajar karena pada Platform Replit, tersedia banyak pembahasan berbasis tutorial mengenai pengenalan dan penggunaan banyak bahasa pemrograman, yang mana fitur seperti ini akan sangat efektif digunakan oleh mereka yang ingin belajar suatu bahasa pemrograman. Karena selain mempelajari teorinya, mereka juga dapat langsung mempraktekannya pada platform Replit tanpa harus menginstall aplikasi pendukung untuk bahasa pemrograman tersebut terlebih dahulu.

Sementara untuk media mengajar, para instruktur atau pengajar dapat menggunakan Replit untuk mengoreksi

pekerjaan anak didiknya melalui komen yang dapat disematkan pada kode mereka, atau bahkan menggunakan platform Replit ini untuk berdiskusi dengan anak didiknya melalui *in-line threads* yang disediakan Replit.

4.3 Alasan Menggunakan Replit

Untuk alasan menggunakan Replit, tentunya alasan utama penulis menggunakan *Replit* karena platform ini dapat digunakan secara gratis. Memang benar jika Replit ini menyediakan paket berbayar, namun jika hanya digunakan sebagai sarana untuk mengembangkan aplikasi berskala kecil, paket gratis Replit ini dirasa sudah cukup atau bahkan sangat mumpuni digunakan.

Selain alasan utama tersebut, alasan penulis menggunakan Replit adalah sebagai berikut :

- **Tidak membutuhkan perangkat pribadi untuk menulis, menjalankan, dan mendeploy kode**

Semua aktivitas pengembangan, baik mulai dari proses debugging, hingga *mendeploy* kode, semuanya dilakukan pada perangkat yang telah disediakan oleh Replit dan tanpa harus melakukan konfigurasi tambahan.

Sebenarnya juga terdapat aplikasi cloud-service serupa dengan Replit, yaitu Cloud9 dari Amazon Web Services(AWS), namun untuk menggunakan layanan ini kita diharuskan untuk membuat *instance/virtual machine* (berbayar), kemudian kita masih diharuskan untuk melakukan konfigurasi lebih lanjut untuk *install* aplikasi Cloud9 dan web server agar dapat melakukan proses debugging hingga *deploying* kode.

- **Terdapat Fitur untuk melakukan kolaborasi secara *real-time***

Dalam pengembangan suatu aplikasi, tentunya akan semakin cepat jika dikerjakan bersama, jadi fitur kolaborasi *real-time* yang ada pada Replit ini akan sangat membantu dalam membangun suatu aplikasi yang membutuhkan kerja sama tim.

Meskipun saat ini sudah terdapat beberapa desktop IDE yang mendukung fitur kolaborasi real-time, misal disini kita sebutkan salah satu contohnya yaitu VSCODE. Namun dengan menggunakan aplikasi semacam ini, biasanya kita akan mengalami masalah kompatibilitas pada perangkat rekan atau tim kita yang menggunakan sistem operasi selain yang didukung oleh pengembang aplikasi.

- **Mendukung untuk kegiatan *full stack development***

Untuk membangun sebuah website (terutama website dinamis), tentunya dibutuhkan dukungan bahasa yang kompleks, yaitu pada bagian front-end, back-end, dan juga database.

Hal tersebut lah yang dapat menjadikan Replit sebagai pilihan yang cocok untuk dapat membangun sebuah website dinamis, hal ini dikarenakan Replit memiliki dukungan terhadap bahasa *front-end*(HTML,CSS, JavaScript), back-end(PHP, Python, JavaScript), dan juga database(SQL).

Sebenarnya juga terdapat *cloud-based* IDE populer yang mendukung untuk kegiatan *full stack development*, contohnya adalah CodeSanbox. Namun pada CodeSanbox, bahasa *back-end* yang didukung hanyalah JavaScript, jadi kita tidak akan bisa menggunakannya jika kita ingin mengembangkan suatu website yang menggunakan PHP sebagai bahasa *back-end*nya.

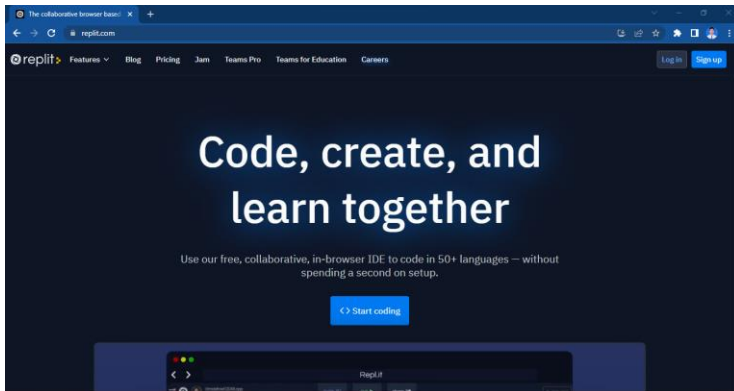
BAB 5

MEMULAI MENGGUNAKAN REPLIT

5.1 Mendaftar

Sebelum dapat memulai menggunakan Replit, hal pertama yang harus dilakukan adalah melakukan pendaftaran pada laman resminya. Untuk tata caranya, berikut adalah langkah-langkah yang dapat dilakukan untuk melakukan pendaftaran pada *platform* Replit.

1. Silakan buka atau akses halaman resmi Replit yang beralamatkan pada *url* berikut (<https://replit.com/>)



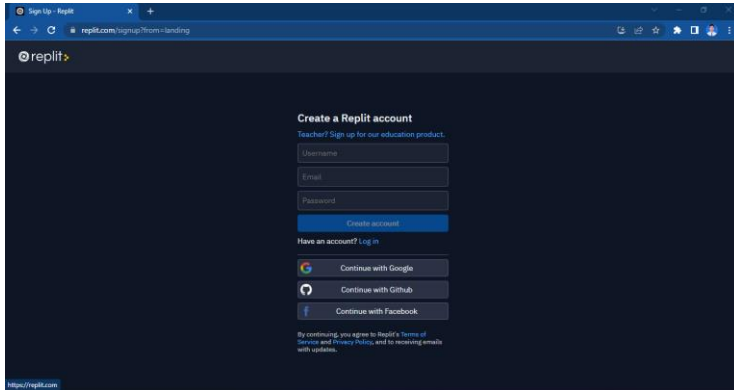
Gambar 5. 1 Halaman Utama Replit

2. Selanjutnya, silakan klik tombol *Signup*/Daftar, tombol ini biasanya berada di bagian pojok kanan atas, pada halaman utama Replit.



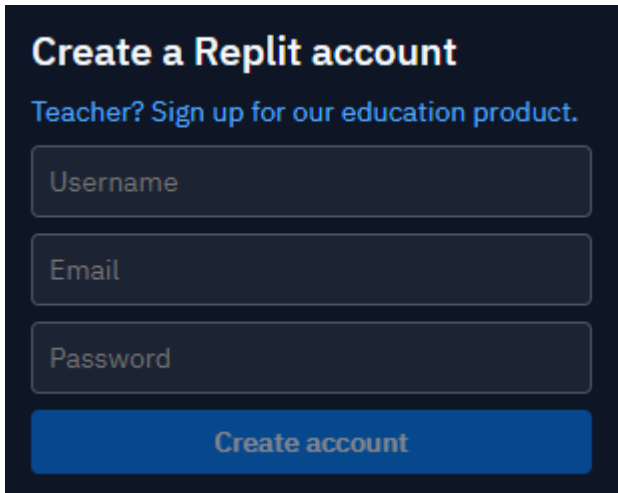
Gambar 5. 2 Tombol Signup Replit

Setelah mengklik tombol tersebut, maka kita akan diarahkan ke halaman pendaftaran Replit seperti pada gambar berikut.



Gambar 5. 3 Halaman Pendaftaran Replit

3. Silakan isi form pendaftaran yang disediakan dengan data yang ingin kita daftarkan, seperti data username, email, dan password. Jika sudah, selanjutnya silakan klik tombol “Create Account”.

The image shows a dark-themed user interface for creating a Replit account. At the top, the text "Create a Replit account" is displayed in a large, bold, white font. Below this, a link "Teacher? Sign up for our education product." is shown in a smaller, blue font. The form consists of three stacked input fields with light blue borders and placeholder text: "Username", "Email", and "Password". At the bottom of the form is a prominent blue button with the white text "Create account".

Create a Replit account

Teacher? Sign up for our education product.

Username

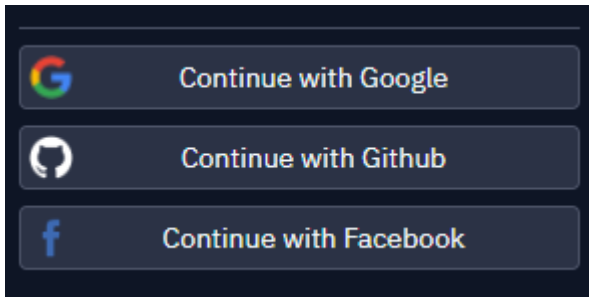
Email

Password

Create account

Gambar 5. 4 Form Pendaftaran Replit

(Opsional), Jika kalian telah memiliki akun pada salah satu layanan berikut, kalian dapat langsung melanjutkan menggunakan akun pada layanan tersebut, tanpa harus mengisi form di atas.



Gambar 5. 5 Tombol Pendaftaran Replit Melalui Layanan Populer

4. Setelah berhasil mendaftar, saat pertama kali login ke halaman Replit, kalian akan diminta untuk mengisi form *profile* dan memilih tema replit yang ingin digunakan, namun isian pada form ini sifatnya opsional, jadi kalian dapat mengisinya hanya jika merasa dibutuhkan. Jika sudah, silakan klik tombol “looks good!”.

🔗 Welcome to Replit. Let's get you set up.

First, who do you want to be on Replit?

Display name Optional

19-121


Short bio/description Optional

A sentence or two about you

Profile picture Optional

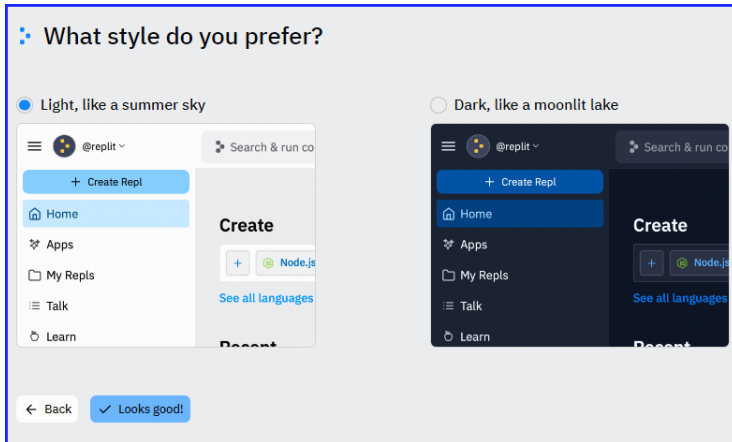
📁 Upload a photo, or drag & drop

Here's how you'll look:

 19-121
@19-121Bihubbi1

✓ Looks good!

Gambar 5. 6 Form Penyiapan Profil Replit



Gambar 5. 7 Pemilihan Tema Replit

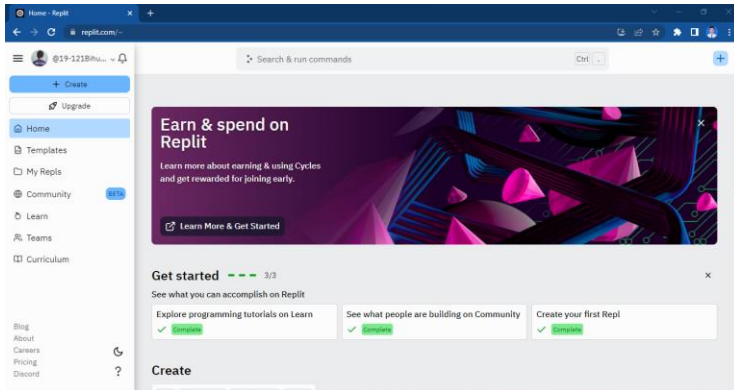
5.2 Membuat Project Baru

Setelah berhasil melakukan pendaftaran pada platform Replit, barulah selanjutnya kita dapat menggunakan fitur utama Replit, yaitu sebagai IDE(integrated development environment) suatu bahasa pemrograman.

Agar dapat menggunakan Replit sebagai IDE, hal yang harus kita lakukan adalah membuat sebuah project atau Repls(istilah project pada Replit) baru. Untuk dapat membuat project atau Repls baru, berikut adalah langkah-langkah yang dapat dilakukan.

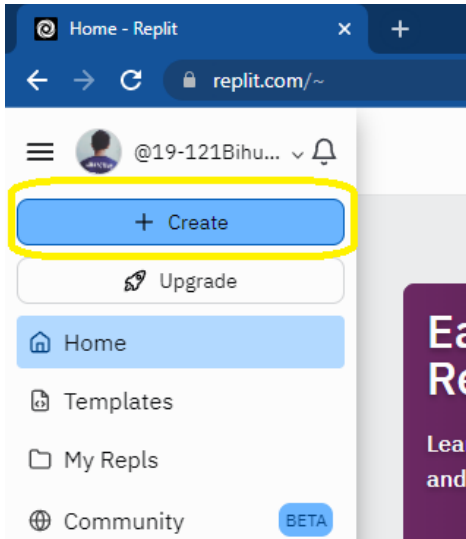
1. Silakan buka halaman *dashboard* pengguna pada alamat berikut (<https://replit.com/~>).

*catatan : Pastikan Anda telah berhasil melakukan *login* pada platform Replit agar dapat mengakses *dashboard* pengguna.



Gambar 5. 8 Halaman Dashboard Replit

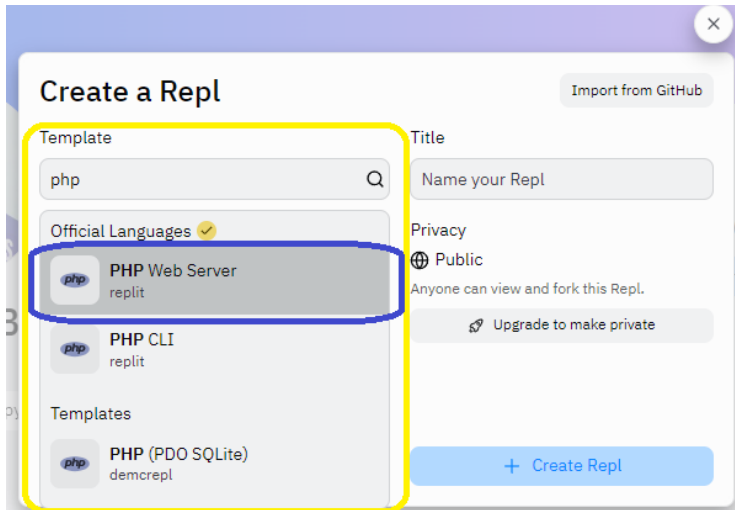
2. Kemudian silakan klik tombol *create* yang ada pada bagian *sidebar menu*.



Gambar 5. 9 Tombol Create Project / Repl

3. Setelah mengklik tombol create, maka akan muncul popup penyiapan project atau repl.

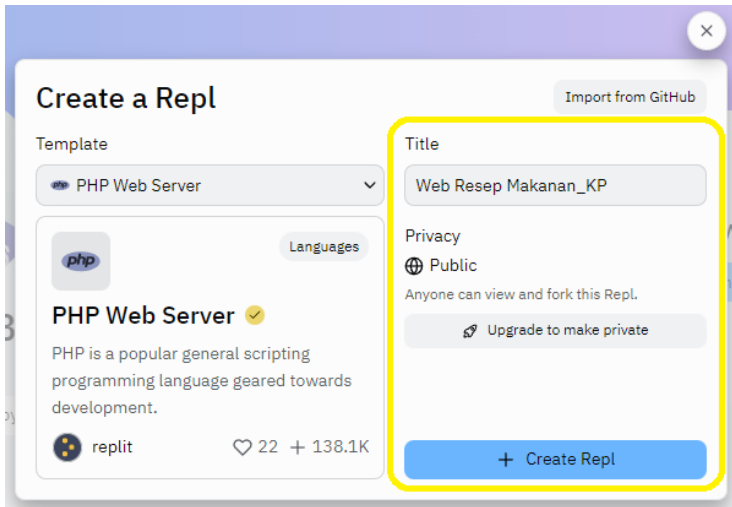
Selanjutnya, silakan pilih bahasa pemrograman sesuai dengan project yang akan dibuat. Misal jika kita ingin membuat project berbasis PHP, maka silakan pilih “PHP Web Server”.



Gambar 5. 10 Form Pemilihan Bahasa Pemrograman

Setelah memilih bahasa pemrograman yang ingin digunakan, selanjutnya silakan beri nama project tersebut, sebagai contoh, disini kita akan memberikan project tersebut dengan nama “Web Resep Makanan_KP”. Jika sudah, silakan klik tombol *Create Repl* untuk membuat *project*.

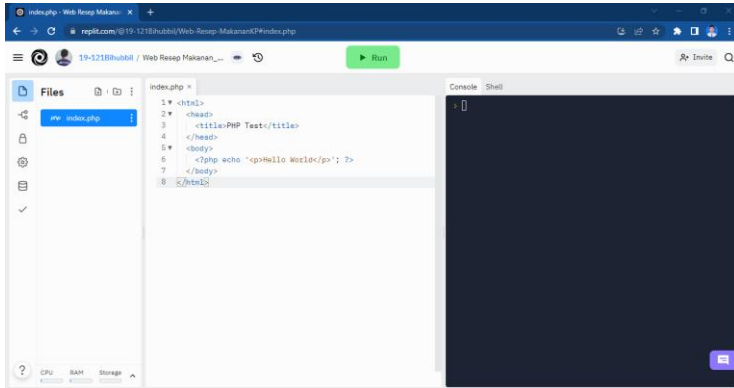
*Catatan : Agar dapat mengubah setelah privasi *project* menjadi *private*, kita perlu melakukan *upgrade plan* ke tingkat yang lebih tinggi (berbayar).



Gambar 5. 11 Form Pemberian Nama dan Pemilihan Setelah Privasi Repl

4. Jika kita telah berhasil melakukan penyiapan *project* atau repl, maka selanjutnya akan diarahkan ke halaman direktori *project* kita, yang mana di dalamnya sudah terdapat fungsi IDE dan *file manager* yang dapat kita

gunakan untuk mengelola *project* yang telah kita buat tersebut.



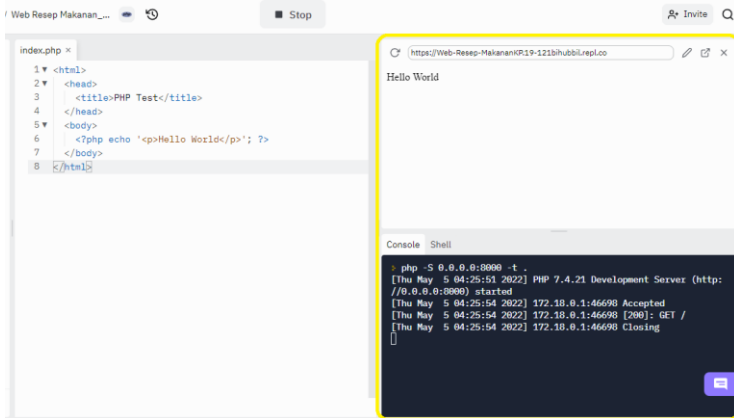
Gambar 5. 12 Halaman IDE Project/Repl Replit

5. Terakhir, agar dapat menjalankan project yang telah Anda buat, silakan klik tombol Run yang berada pada topbar menu.



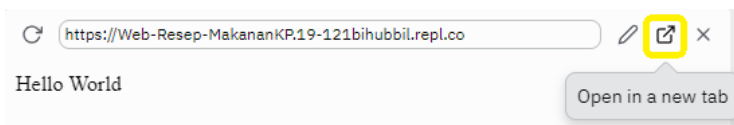
Gambar 5. 13 Tombol Run Project/Repl

Karena project yang kita buat menggunakan bahasa pemrograman PHP, maka sistem Replit akan menjalankan fungsi *PHP Development Server* pada *subdomain* khusus yang mengarah pada direktori project kita. Dan jika tidak ada kesalahan pada project yang kita jalankan, maka preview dari project yang dijalankan akan ditampilkan pada mini peramban yang terbentuk diatas *console*.



5. 14 Console Project/Repl Replit

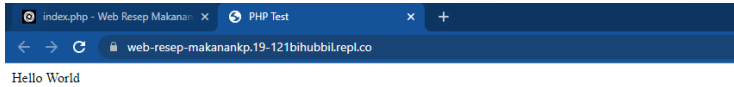
Jika dirasa mini peramban yang disediakan Replit terlalu kecil atau tidak sesuai dengan ukuran layar dari project yang akan dibuat. Kita dapat mengklik tombol *Open in a new tab* yang terletak pada bagian atas mini peramban.



Gambar 5. 15 Tombol Open In New Tab Mini Browser Replit

Maka alamat url dari subdomain yang mengarah ke project kita akan terbuka pada *tab* baru di peramban yang sedang

kita gunakan seperti pada gambar berikut. Selanjutnya, kalian tinggal mengatur ukurannya menggunakan *developer tools* yang ada peramban kalian.



Gambar 5. 16 Tampilan Output Project/Repl Pada Tab Baru

BAB 6

IMPLEMENTASI PENGGUNAAN REPLIT PADA PEMBUATAN WEBSITE (STUDI KASUS : WEBSITE RESEP MAKANAN)

6.1 Konsep Aplikasi(Website)

6.1.1 Gambaran Umum

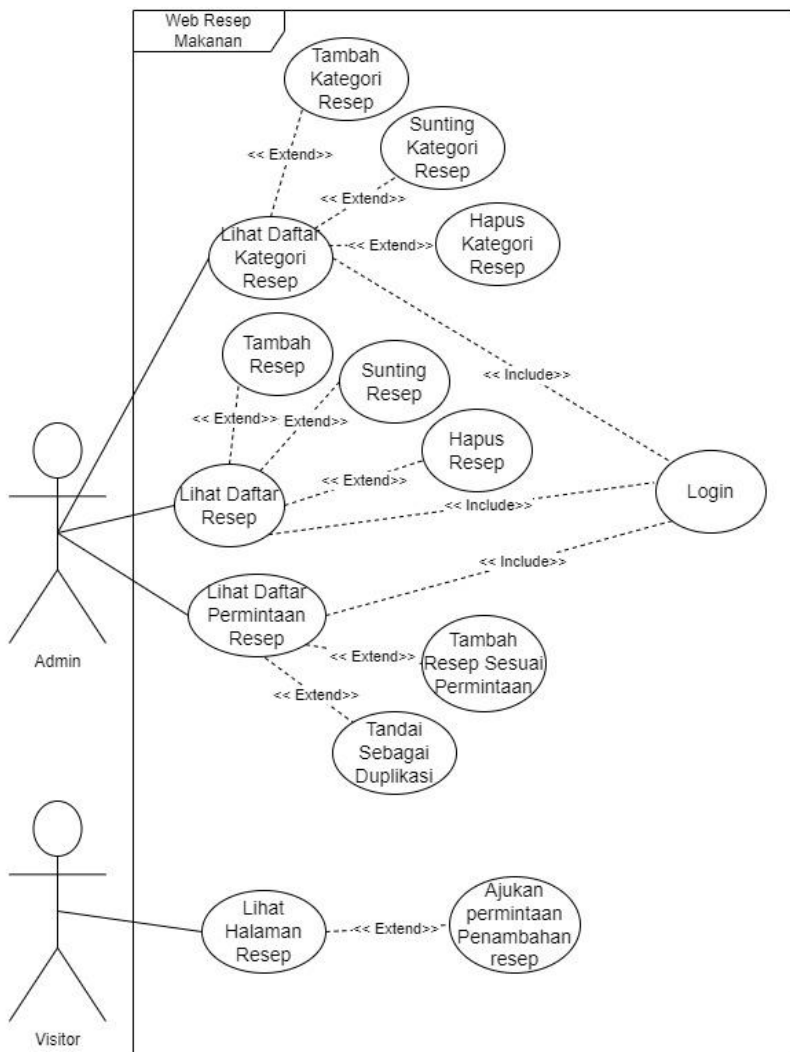
Untuk gambaran umumnya, aplikasi yang akan kita coba buat untuk mengimplementasikan penggunaan Replit pada pemrograman web, konsepnya adalah seperti sebuah blog resep makanan.

Jadi pada blog tersebut nantinya akan terdapat sistem yang dapat digunakan untuk mengelola konten yang terdapat didalamnya. Sebagai contoh, aplikasi ini nantinya akan dapat melakukan penambahan resep, penyuntingan resep, serta penghapusan resep. Selain itu pada aplikasi ini juga akan terdapat fitur untuk menerima permintaan resep yang dapat

diajukan oleh *visitor*, yang mana dari permintaan tersebut, nantinya *admin* akan dapat menambahkannya kedalam database resep aplikasi, sehingga akan dapat tampil pada blog.

6.1.2 Use Case

Agar dapat memahaminya secara lebih detil mengenai penggunaan aplikasi yang akan dibuat, berikut saya sertakan *user case* diagram beserta detil aktivitas yang dapat dilakukan oleh tiap aktornya.



Gambar 6. 1 Use Case

Detil aktivitas tiap aktor :

Visitor

- Melihat Halaman Resep

Untuk melihat halaman resep, visitor atau admin dapat memanfaatkan daftar/katalog resep yang ada di halaman utama untuk memilih makanan yang ingin dilihat resepnya, sehingga nantinya akan diarahkan ke halaman resep.

- Melakukan Permintaan Penambahan Resep Makanan

Untuk melakukan permintaan penambahan resep makanan, visitor dapat memilih menu request resep yang ada pada halaman utama maupun halaman resep, sehingga nantinya akan diarahkan ke halaman form permintaan penambahan resep makanan. Selanjutnya visitor diharuskan untuk mengisi form tersebut agar permintaannya dapat dikirimkan.

Admin :

- Melakukan Penambahan Kategori Resep

Untuk melakukan penambahan kategori resep, admin dapat memilih menu tambah kategori resep yang ada pada halaman dashboard admin, sehingga nantinya akan diarahkan ke halaman form penambahan kategori resep. Selanjutnya admin diharuskan mengisi form tersebut agar dapat menambahkan kategori resep ke dalam database.

- Melihat Daftar Kategori Resep

Untuk melihat daftar kategori resep, admin dapat memilih menu daftar kategori resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar kategori resep yang telah ditambahkan sebelumnya.

- Melakukan Penyuntingan Kategori Resep

Untuk melakukan penyuntingan kategori resep, admin dapat memilih menu daftar kategori resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar kategori resep yang telah ditambahkan sebelumnya. Selanjutnya admin dapat mengklik tombol

sunting yang ada pada setiap item di daftar resep, sesuai dengan kategori resep mana yang ingin disunting, sehingga nantinya akan diarahkan kembali ke halaman form sunting kategori resep. Kemudian admin dapat melakukan perubahan pada pada form tersebut dan menyimpan perubahannya, agar nantinya data pada database sesuai kategori resep yang ingin disunting juga dapat berubah.

- Melakukan Penghapusan Kategori Resep

Untuk melakukan penghapusan kategori resep, admin dapat memilih menu daftar kategori resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar kategori resep yang telah ditambahkan sebelumnya. Selanjutnya admin dapat mengklik tombol hapus yang ada pada setiap item di daftar kategori resep, sesuai dengan kategori resep mana yang ingin dihapus, agar nantinya data kategori resep yang dipilih dapat terhapus.

- Melakukan Penambahan Resep

Untuk melakukan penambahan resep, admin dapat memilih menu tambah resep yang ada pada halaman dashboard admin, sehingga nantinya akan diarahkan ke halaman form penambahan resep. Selanjutnya admin diharuskan mengisi form tersebut agar dapat menambahkan resep ke dalam database.

- Melihat Daftar Resep

Untuk melihat daftar resep, admin dapat memilih menu daftar resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar resep yang telah ditambahkan sebelumnya.

- Melakukan Penyuntingan Resep

Untuk melakukan penyuntingan resep, admin dapat memilih menu daftar resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar resep yang telah ditambahkan sebelumnya. Selanjutnya admin dapat mengklik tombol sunting yang ada pada setiap item di daftar resep, sesuai dengan resep

mana yang ingin disunting, sehingga nantinya akan diarahkan kembali ke halaman form sunting resep. Kemudian admin dapat melakukan perubahan pada form tersebut dan menyimpan perubahannya, agar nantinya data pada database sesuai resep yang ingin disunting juga dapat berubah.

- Melakukan Penghapusan Resep

Untuk melakukan penghapusan resep, admin dapat memilih menu daftar resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar resep yang telah ditambahkan sebelumnya. Selanjutnya admin dapat mengklik tombol hapus yang ada pada setiap item di daftar resep, sesuai dengan resep mana yang ingin dihapus, agar nantinya data resep yang dipilih dapat terhapus.

- Melihat Daftar Permintaan Resep

Untuk melihat daftar permintaan resep, admin dapat memilih menu daftar permintaan resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke

halaman daftar permintaan resep yang telah diajukan sebelumnya oleh visitor.

- Melakukan Penambahan Resep Sesuai Permintaan Resep

Untuk dapat melakukan penambahan resep sesuai dengan permintaan resep, admin dapat memilih menu daftar permintaan resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar permintaan resep yang telah diajukan sebelumnya oleh visitor. Selanjutnya admin dapat mengklik tombol tambah yang ada pada setiap item di daftar permintaan resep, sesuai dengan data permintaan resep mana yang ingin ditambahkan, sehingga nantinya akan diarahkan ke halaman form penambahan resep, dengan data nama resep yang otomatis terisi sesuai dengan data nama resep pada permintaan resep yang dipilih. Kemudian admin dapat mengisi bagian form lainnya sesuai dengan catatan yang ada pada permintaan resep, agar nantinya data dari permintaan resep dapat ditambahkan sebagai resep baru dalam database.

- Menandai Permintaan Resep Sebagai Duplikasi

Untuk dapat menandai permintaan resep sebagai duplikasi, admin dapat memilih menu daftar permintaan resep yang ada pada dashboard admin, sehingga nantinya akan diarahkan ke halaman daftar permintaan resep yang telah diajukan sebelumnya oleh visitor. Selanjutnya admin dapat mengklik tombol duplikat yang ada pada setiap item di daftar permintaan resep, sesuai dengan data permintaan resep mana yang ingin ditandai sebagai duplikat, agar nantinya status dari permintaan resep menjadi duplikat dan tidak lagi ditampilkan pada halaman daftar permintaan resep.

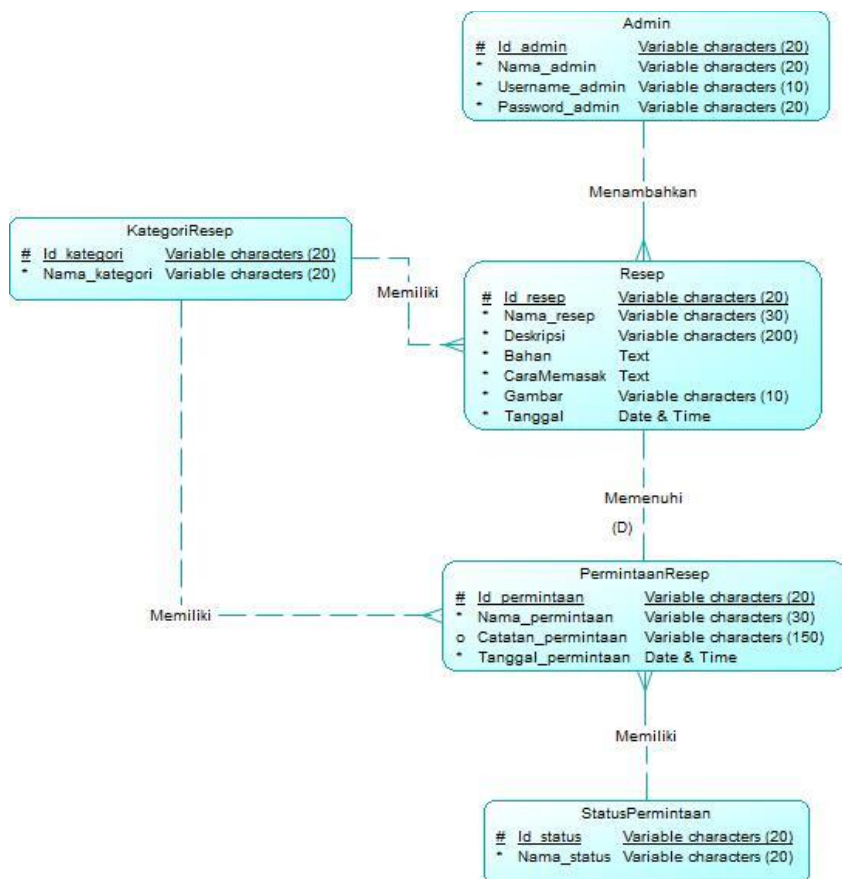
6.2 Perancangan Database

6.2.1 Rancangan Model *Database*

Sebelum membuat dan menerapkan *database* pada web resep makanan, tentunya disini kita harus merancang model *databasenya* terlebih dahulu. Perancangan model *database* ini sendiri berfungsi untuk memudahkan kita dalam mengidentifikasi bagian-bagian apa saja yang diperlukan untuk membuat database agar sesuai dengan kebutuhan sistem yang akan dibuat, contohnya seperti table beserta komponennya, dan juga relasi antar satu table dan table lainnya.

Berikut adalah model *database* yang akan kita terapkan dalam sistem web resep makanan.

- Conceptual Data Model (CDM)



Gambar 6. 2 Conceptual Data Model (CDM)

Dapat terlihat pada conceptual data model di atas, dalam pembuatan website resep makanan, di sini kita membuat 5 entitas yang terdiri dari :

- o Admin
- o Resep
- o Permintaan Resep
- o Kategori Resep
- o Status Permintaan

Sesuai dengan namanya, kelima entitas tersebut nantinya akan berfungsi untuk menyimpan data admin, resep, kategori resep, permintaan resep, serta status permintaan resep pada sistem web resep makanan yang akan dibuat.

Seperti yang kita ketahui sendiri pada suatu sistem, antara entitas satu dengan lainnya sangat mungkin saling berhubungan, hal ini berguna untuk menghubungkan 2 atau lebih data yang saling terkait antara satu dengan lainnya.

Berdasarkan model diatas, berikut adalah hubungan atau *relationship* yang dimiliki tiap entitas :

- o ***Admin* – Resep**

Tiap *admin* dapat menambahkan satu atau banyak resep.

- o ***Resep* – *Admin***

Tiap resep harus ditambahkan oleh satu *admin*.

- o **Kategori Resep – Resep**

Tiap kategori resep dapat memiliki satu atau banyak resep.

- o ***Resep* – Kategori Resep**

Tiap resep dapat memiliki satu kategori resep.

- o **Kategori Resep – Permintaan Resep**

Tiap kategori resep dapat memiliki satu atau banyak permintaan resep.

- o **Permintaan Resep – Kategori Resep**

Tiap permintaan resep dapat memiliki satu kategori resep.

- o ***Resep* – Permintaan Resep**

Tiap resep dapat memenuhi satu permintaan resep.

- o **Permintaan Resep – Resep**

Tiap permintaan resep dapat dipenuhi oleh satu resep.

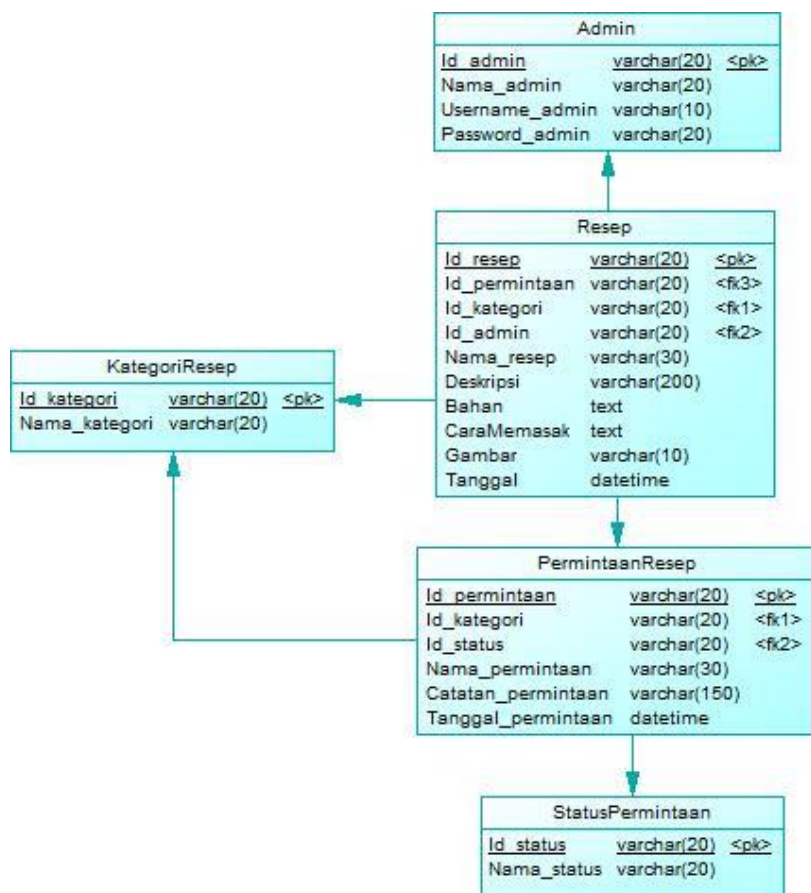
- o **Permintaan Resep – Status Permintaan**

Tiap permintaan resep harus memiliki satu status permintaan.

- o **Status Permintaan – Permintaan Resep**

Tiap status permintaan dapat memiliki satu atau banyak permintaan resep.

- *Physical Data Model (PDM)*



Gambar 6. 3 Physical Data Model (PDM)

Physical Data Model atau PDM sendiri merupakan representasi fisik dari *Conceptual Data Model* (CDM), jadi model inilah yang nantinya akan kita gunakan untuk merancang database sistem website resep makanan. Untuk penerapannya pada sistem database, hal ini akan kita bahas

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

pada section selanjutnya, yaitu 6.2.2 Penerapan Model *Database Pada SQLite*.

6.2.2 Penerapan Model *Database Pada SQLite*

Setelah mengetahui rancangan model *databasenya*, selanjutnya kita dapat menerapkan rancangan model database tersebut pada *database engine* SQLite(SQLite3). Untuk membuat dan mengelola SQLite database, sebenarnya terdapat 2 opsi, yaitu menggunakan modul SQLite3 yang ada pada PHP, dan juga dapat langsung menggunakan perintah SQLite3 jika kita dapat mengakses *console/terminal/shell*.

Karena pada Replit ini kita dapat dan diperbolehkan mengakses *console* dan *shell*, maka untuk perancangan *databasenya*, disini kita akan mempraktekkannya menggunakan *console/shell* yang ada pada Replit. Sementara untuk proses mengelola data pada *databasenya*, barulah nantinya kita akan memanfaatkan *module* SQLite3 yang terdapat pada PHP.

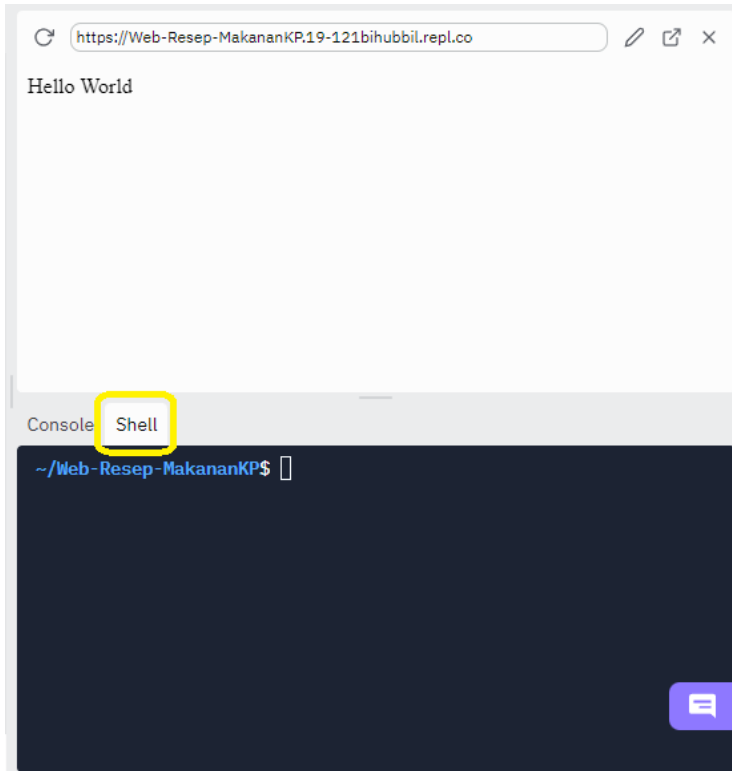
Berikut adalah langkah-langkah yang dapat dilakukan untuk membuat *database* menggunakan SQLite3 pada Replit.

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

*Catatan : Sebelum mengikuti tutorial berikut, pastikan kalian telah mengikuti semua tutorial yang terdapat pada Bab 5 Memulai Menggunakan Replit).

1. Silakan buka halaman IDE project atau repl yang telah kita buat sebelumnya. Selanjutnya, silakan klik tombol *Shell* yang terdapat pada bagian bawah mini peramban.

*Catatan : Sebenarnya kita juga menggunakan *console* untuk menjalankan perintah SQLite3, namun bagian *console* tersebut sebelumnya telah digunakan untuk menjalankan PHP *Development Server*, jadi kita tidak akan bisa menginputkan kembali perintah pada *console*, kecuali kita menghentikan terlebih dahulu perintah yang sedang berjalan.



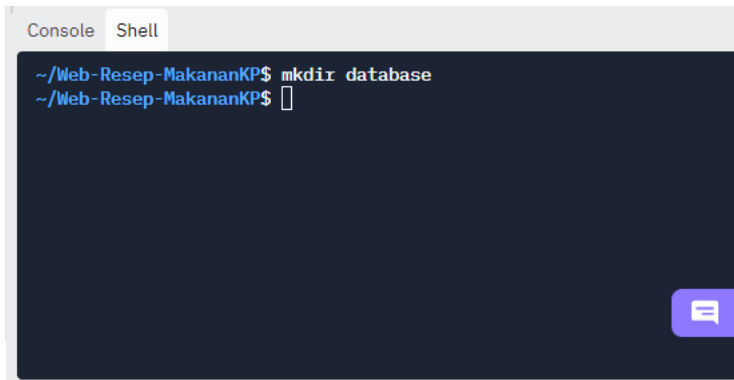
Gambar 6. 4 Shell Project/Repl Replit

2. Pada CLI(*Command-Line Interface*) *Shell*, silakan masukkan perintah berikut untuk membuat direktori baru, yang nantinya akan kita gunakan untuk menyimpan berkas *database* yang akan dibuat.

*Catatan : Langkah ini adalah opsional, karena kalian akan tetap dapat menyimpan berkas *database* pada *home* direktori *project* yg dibuat.

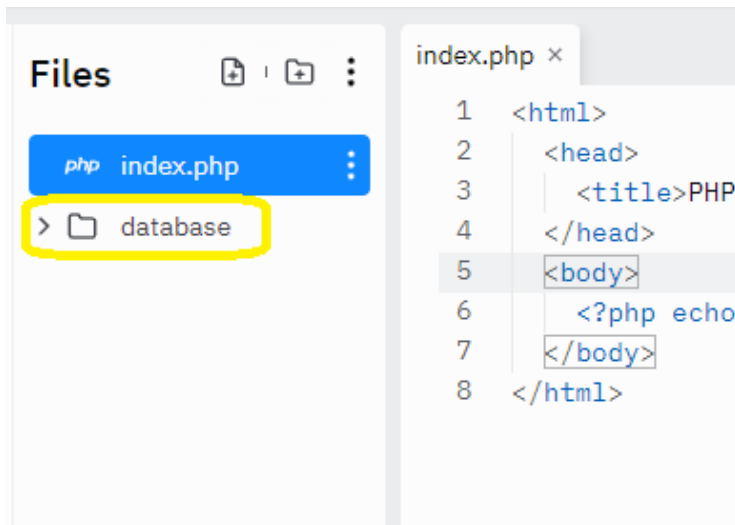
```
$ mkdir nama_direkori_database
```

Tabel 6. 1 Perintah Membuat Direktori Database



Gambar 6. 5 Perintah Membuat Direktori Database

Setelah menjalankan perintah tersebut, maka akan muncul direktori baru pada file manager IDE, sesuai dengan nama direktori yang kita buat.



Gambar 6. 6 Hasil Perintah Membuat Direktori Database (File Manager)

3. Fokus kembali pada CLI(*Command-Line Interface*) Shell, kemudian jalankan perintah berikut untuk memanggil/menjalankan aplikasi SQLite3 berbasis CLI.

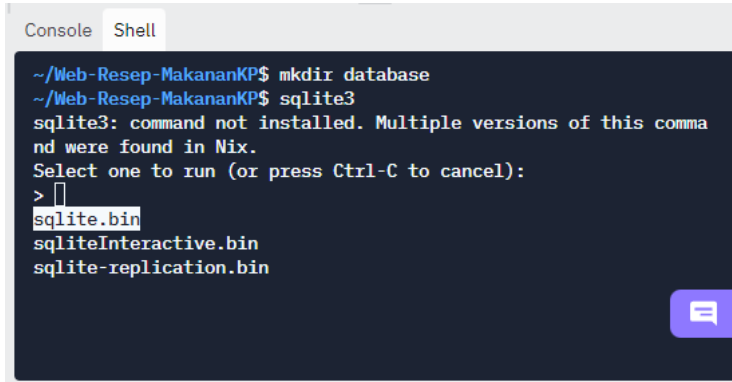
```
$ sqlite3
```

Tabel 6. 2 Perintah Memanggil/Menjalankan Aplikasi SQLite3 Berbasis

CLI

Setelah menjalankan perintah diatas, akan terdapat prompt yang menginstruksikan kita untuk memilih versi dari

SQLite *command*, silakan pilih salah satu versi *command* yang kalian ingin gunakan.



```
~/Web-Resep-MakananKP$ mkdir database
~/Web-Resep-MakananKP$ sqlite3
sqlite3: command not installed. Multiple versions of this command were found in Nix.
Select one to run (or press Ctrl-C to cancel):
> sqlite.bin
sqliteInteractive.bin
sqlite-replication.bin
```

Gambar 6. 7 Tampilan CLI Pemilihan Versi SQLite3

Maka kalian akan berhasil menjalankan SQLite3, dan selanjutnya dapat menjalankan perintah SQLite didalamnya.

```
Console Shell
bash: sqlite: command not found
~/Web-Resep-MakananKP$ sqlite3
sqlite3: command not installed. Multiple versions of this command
were found in Nix.
Select one to run (or press Ctrl-C to cancel):
Add 'sqlite.bin' to replit.nix if you want to install 'sqlite3'
in this repl.
/nix/store/myz08ag3y1f7hq26vc6wbd19m9af9hmd-sqlite-3.35.5-bin
SQLite version 3.35.5 2021-04-19 18:32:05
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> 
```

Gambar 6. 8 Tampilan CLI Setelah Memilih Versi SQLite3

- Setelah itu, silakan jalankan perintah berikut untuk dapat membuat berkas *database* baru pada direktori *project* kita.
*Catatan : Perintah ini sebenarnya adalah perintah untuk membuka suatu berkas database SQLite yang sudah ada, namun jika kita memasukkan nama berkas yang tidak terdapat pada direktori, maka perintah ini akan membuat berkas untuk *database* baru dan kemudian membukanya.

```
sqlite> .open
nama_direkori_database/nama_berkas_datab
ase
```

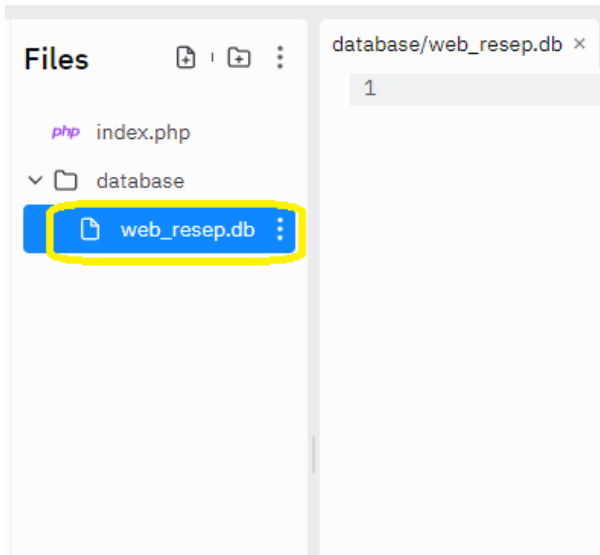
Tabel 6. 3 Perintah Untuk Membuat/Membuka Berkas Database SQLite

```
Console  Shell

~/Web-Resep-MakananKP$ sqlite3
sqlite3: command not installed. Multiple versions of this command were found in Nix.
Select one to run (or press Ctrl-C to cancel):
Add 'sqlite.bin' to replit.nix if you want to install 'sqlite3' in this repl.
/nix/store/myz08ag3y1f7hq26vc6wbd19m9af9hmd-sqlite-3.35.5-bin
SQLite version 3.35.5 2021-04-19 18:32:05
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open database/web_resep.db
sqlite> 
```

Gambar 6. 9 Hasil Perintah Membuat/Membuka Berkas Database SQLite (Shell)

Setelah menjalankan perintah tersebut, maka akan muncul berkas baru pada file manager IDE, sesuai dengan nama berkas yang kita buat.



Gambar 6. 10 Hasil Perintah Membuat/Membuka Berkas Database SQLite (File Manager)

5. Untuk selanjutnya adalah tahap pembuatan *table* (beserta *column* dan relasinya) pada database, sesuai dengan model yang telah dibuat sebelumnya, yaitu pada Gambar 6. 3 Pysical Data Model (PDM) .

o **Pembuatan *table* ADMIN**

Pada model, *table* ini memiliki kolom :

ADMIN	
ID_ADMIN varchar(20)	Primary Key
NAMA_ADMIN varchar(20)	
USERNAME_ADMIN varchar(10),	
PASSWORD_ADMIN varchar(20)	

Tabel 6. 4 Konsep Table ADMIN

Untuk dapat membuat table ADMIN pada SQLite, silakan gunakan perintah sql “create table” seperti berikut.

```
sqlite> CREATE TABLE "ADMIN" (  
    "ID_ADMIN" varchar(20) NOT NULL,  
    "NAMA_ADMIN" varchar(20) NOT  
NULL,  
    "USERNAME_ADMIN" varchar(10) NOT  
NULL,  
    "PASSWORD_ADMIN" varchar(20) NOT  
NULL,  
    PRIMARY KEY ("ID_ADMIN")  
);
```

Tabel 6. 5 Perintah SQL Untuk Membuat Table Admin

o **Pembuatan *table* KATEGORIRESEP**

Pada model, *table* ini memiliki kolom :

KATEGORIRESEP	
ID_KATEGORI varchar(20)	Primary Key
NAMA_KATEGORI varchar(20)	

Tabel 6. 6 Konsep Table KATEGORIRESEP

Untuk dapat membuat table KATEGORIRESEP pada SQLite, silakan gunakan perintah sql “create table” seperti berikut.

```
sqlite> CREATE TABLE "KATEGORIRESEP" (  
    "ID_KATEGORI"      varchar(20) NOT  
    NULL,  
    "NAMA_KATEGORI"    varchar(20) NOT  
    NULL,  
    PRIMARY KEY ("ID_KATEGORI")  
);
```

Tabel 6. 7 Perintah SQL Untuk Membuat Table Kategori Resep

o **Pembuatan *table* STATUSPERMINTAAN**

Pada model, *table* ini memiliki kolom :

STATUSPERMINTAAN	
ID_STATUS varchar(20)	Primary Key
NAMA_STATUS varchar(20)	

Tabel 6. 8 Konsep Table STATUSPERMINTAAN

Untuk dapat membuat table STATUSPERMINTAAN pada SQLite, silakan gunakan perintah sql “create table” seperti berikut.

```
sqlite> CREATE TABLE "STATUSPERMINTAAN"
(
    "ID_STATUS" varchar(20) NOT NULL,
    "NAMA_STATUS"      varchar(20) NOT
NULL,
    PRIMARY KEY ("ID_STATUS")
);
```

Tabel 6. 9 Perintah SQL Untuk Membuat Table Status Permintaan

o **Pembuatan *table* RESEP**

Pada model, *table* ini memiliki kolom :

RESEP	
ID_RESEP varchar(20)	Primary Key
ID_PERMINTAAN varchar(20)	Foreign Key
ID_KATEGORI varchar(20)	Foreign Key
ID_ADMIN varchar(20)	Foreign Key
NAMA_RESEP varchar(30)	
DESKRIPSI varchar(200)	
BAHAN text	
CARAMEMASAK text	
GAMBAR varchar(10)	
TANGGAL datetime	

Tabel 6. 10 Konsep Table RESEP

Sementara untuk relasinya, table ini memiliki relasi ke table :

- ADMIN, dengan referensi kolom (ID_ADMIN)
- KATEGORI RESEP, dengan referensi kolom (ID_KATEGORI)

- PERMINTAANRESEP, dengan referensi kolom (ID_PERMINTAAN).

Untuk dapat membuat *table* dengan kriteria tersebut pada SQLite, kita dapat menggunakan perintah sql “create table” seperti berikut.

```
sqlite> CREATE TABLE "RESEP" (  
    "ID_RESEP" varchar(20) NOT NULL,  
    "ID_PERMINTAAN" varchar(20),  
    "ID_KATEGORI" varchar(20),  
    "ID_ADMIN" varchar(20) NOT NULL,  
    "NAMA_RESEP" varchar(30) NOT  
NULL,  
    "DESKRIPSI" varchar(200) NOT NULL,  
    "BAHAN" text NOT NULL,  
    "CARAMEMASAK" text NOT NULL,  
    "GAMBAR" varchar(10) NOT NULL,  
    "TANGGAL" datetime NOT NULL,  
    PRIMARY KEY("ID_RESEP"),  
    FOREIGN KEY("ID_PERMINTAAN")  
REFERENCES  
"PERMINTAANRESEP"("ID_PERMINTAAN") on  
delete set null on update cascade,  
    FOREIGN KEY("ID_KATEGORI")  
REFERENCES  
"KATEGORIRESEP"("ID_KATEGORI") on delete  
set null on update cascade,  
    FOREIGN KEY("ID_ADMIN") REFERENCES  
"ADMIN"("ID_ADMIN") on delete restrict  
on update cascade  
);
```

Tabel 6. 11 Perintah SQL Untuk Membuat Table Resep

o **Pembuatan *table* PERMINTAANRESEP**

Pada model, *table* ini memiliki kolom :

PERMINTAANRESEP	
ID_PERMINTAAN varchar(20)	Primary Key
ID_KATEGORI varchar(20)	Foreign Key
ID_STATUS varchar(20)	Foreign Key
NAMA_PERMINTAAN varchar(30)	
NAMA_PERMINTAAN varchar(30)	
CATATAN_PERMINTAAN varchar(150)	
TANGGAL_PERMINTAAN datetime	

Tabel 6. 12 Konsep Table PERMINTAANRESEP

Sementara untuk relasinya, table ini memiliki relasi ke table :

- ADMIN, dengan referensi kolom (ID_ADMIN)
- KATEGORI, dengan referensi kolom (ID_KATEGORI)

Untuk dapat membuat table PERMINTAANRESEP pada SQLite, silakan gunakan perintah sql “create table” seperti berikut.

```
sqlite> CREATE TABLE "PERMINTAANRESEP" (  
    "ID_PERMINTAAN"    varchar(20) NOT  
NULL,  
    "ID_KATEGORI"      varchar(20),  
    "ID_STATUS"        varchar(20),  
    "NAMA_PERMINTAAN"  varchar(30) NOT  
NULL,  
    "CATATAN_PERMINTAAN"  
        varchar(150),  
    "TANGGAL_PERMINTAAN"    datetime  
NOT NULL,  
    PRIMARY KEY("ID_PERMINTAAN"),  
    FOREIGN KEY("ID_KATEGORI")  
REFERENCES  
"KATEGORI"("ID_KATEGORI") on delete  
set null on update cascade,  
    FOREIGN KEY("ID_STATUS")  
REFERENCES  
"STATUSPERMINTAAN"("ID_STATUS") on  
delete restrict on update cascade  
);
```

Tabel 6. 13 Perintah SQL Untuk Membuat Table
Permintaan Resep

6. Terakhir, untuk memastikan semua perubahan tersimpan, silakan simpan berkas database sebelumnya ke berkas database baru. Misal disini kita akan menyimpannya ke file baru dengan nama `webresep.db`, maka perintahnya adalah seperti berikut.

```
sqlite> .save webresep.db
```

Tabel 6. 14 Perintah Untuk Menyimpan Hasil Perubahan Database
Pada Berkas Baru

Hal tersebut dilakukan guna menghindari masalah “*malformed database disk image*” yang mungkin disebabkan karena permasalahan sinkronisasi *system* Replit ke berkas database yang sedang *diedit*, seperti pada gambar berikut.

```
sqlite> .open database/web_resep.db
sqlite> .tables
Error: database disk image is malformed
sqlite> 
```

Gambar 6. 11 Tampilan CLI Error “Malformed Database Disk Image”

6.3 Perancangan Interface

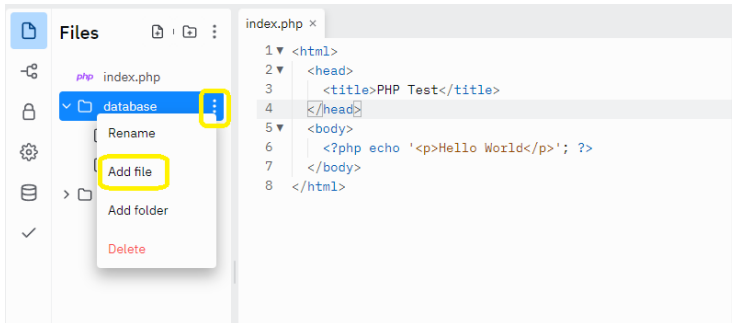
6.3.1 Pembuatan Modul Koneksi *Database*

Sebelum membuat rancangan interface dari website yang akan dibuat. Pada tahap ini kita akan terlebih dahulu membuat suatu modul yang nantinya akan dapat digunakan untuk menghubungkan dan mengelola *database* melalui interface.

Untuk langkah-langkahnya pembuatannya adalah sebagai berikut, asumsikan kita akan membuat modul tersebut pada direktori yang sama dengan tempat menyimpan berkas *database* sebelumnya, yaitu pada direktori *database* :

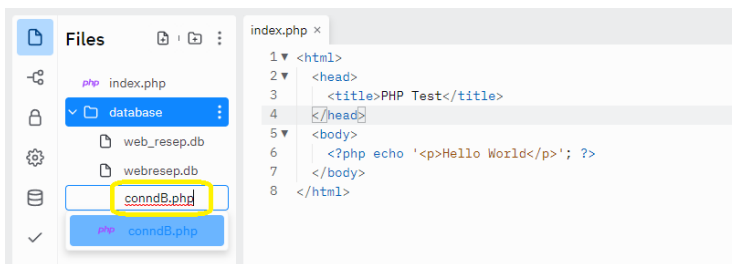
*Catatan : Untuk tutorial pembuatan berkas *database*, kalian dapat melihatnya pada section 6.2.2 Penerapan Model *Database* Pada *SQLite*

1. Pada bagian file manager IDE, silakan klik 3 titik yang berada di sebelah kanan direktori *database*, kemudian pilih *Add file*.



Gambar 6. 12 Cara Membuat/Menambahkan Berkas Baru Menggunakan File Manager

kemudian, silakan beri nama dan ekstensi dari berkas modul yang ingin dibuat, misal disini kita akan memberi nama berkas tersebut conndB dengan ekstensi php. Jika sudah, silakan klik enter untuk menambahkan berkas tersebut.



Gambar 6. 13 Membuat Berkas conndB.php

2. Setelah dibuat, maka selanjutnya berkas tersebut akan otomatis terbuka pada *file editor* IDE. Pada bagian *file*

editor berkas tersebut, kemudian silakan masukkan kode berikut.

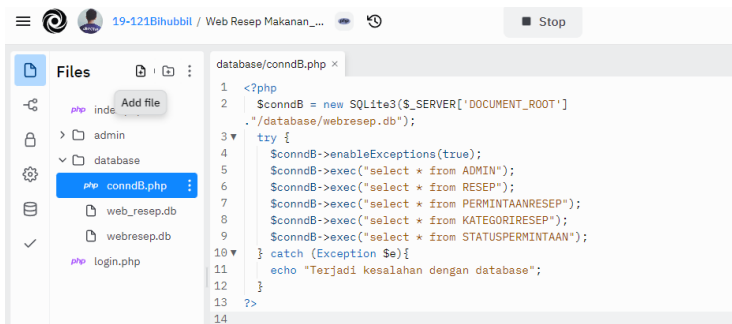
conndB.php
<pre><?php \$conndB = new SQLite3(\$_SERVER['DOCUMENT_ROOT'] ."/database/webresep.db"); try { \$conndB->enableExceptions(true); \$conndB->exec("select * from ADMIN"); \$conndB->exec("select * from RESEP"); \$conndB->exec("select * from PERMINTAANRESEP"); \$conndB->exec("select * from KATEGORI RESEP"); \$conndB->exec("select * from STATUSPERMINTAAN"); } catch (Exception \$e){ echo "Terjadi kesalahan dengan database"; } ?></pre>

Tabel 6. 15 Kode Untuk Berkas conndB.php

Keterangan :

Kode tersebut adalah kode yang berisi fungsi untuk memanggil salah satu module PHP yaitu SQLite3, dan mendefinisikannya pada sebuah variable. Kemudian dilakukan pengecekan semua table yang telah dibuat sebelumnya guna memastikan database yang diakses benar dan tidak ada *table* yang kurang.

Dan berikut adalah tampilannya pada *file editor* Replit



The screenshot shows the Replit interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named '19-121Bihubbl / Web Resep Makanan...' with a 'database' folder containing 'conndB.php', 'web_resep.db', and 'webresep.db'. The code editor shows the contents of 'database/conndB.php'.

```
1 <?php
2 $conndB = new SQLite3($_SERVER['DOCUMENT_ROOT']
3     . "/database/webresep.db");
4 try {
5     $conndB->enableExceptions(true);
6     $conndB->exec("select * from ADMIN");
7     $conndB->exec("select * from RESEP");
8     $conndB->exec("select * from PERMINTAANRESEP");
9     $conndB->exec("select * from KATEGORIRESAP");
10    $conndB->exec("select * from STATUSPERMINTAAN");
11 } catch (Exception $e){
12     echo "Terjadi kesalahan dengan database";
13 }
14 ?>
```

6. 14 Tampilan Kode Pada Berkas conndB.php (Code Editor)

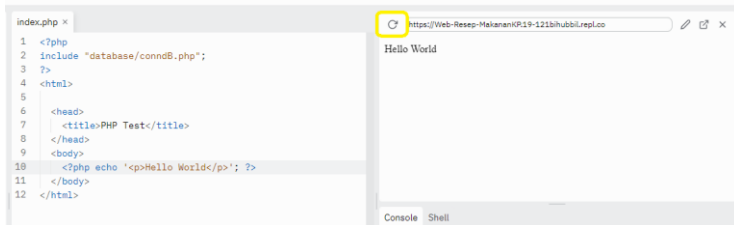
3. Untuk dapat memastikan apakah terdapat error atau tidak pada kode tersebut, silakan buka *file* index.php, kemudian tambahkan kode berikut pada bagian awal.

index.php

```
<?php
include "database/conndB.php";
?>
```

Tabel 6. 16 Kode Untuk Berkas *index.php* (Uji Coba Kode Pada Berkas *conndB.php*)

Selanjutnya silakanklik tombol refresh yang ada pada mini peramban, jika tidak terdapat *error* maka tampilannya akan sama seperti sebelumnya, yaitu hanya terdapat tulisan *Hello World* seperti pada gambar berikut.



Gambar 6. 15 Tombol Refresh Mini Browser Replit

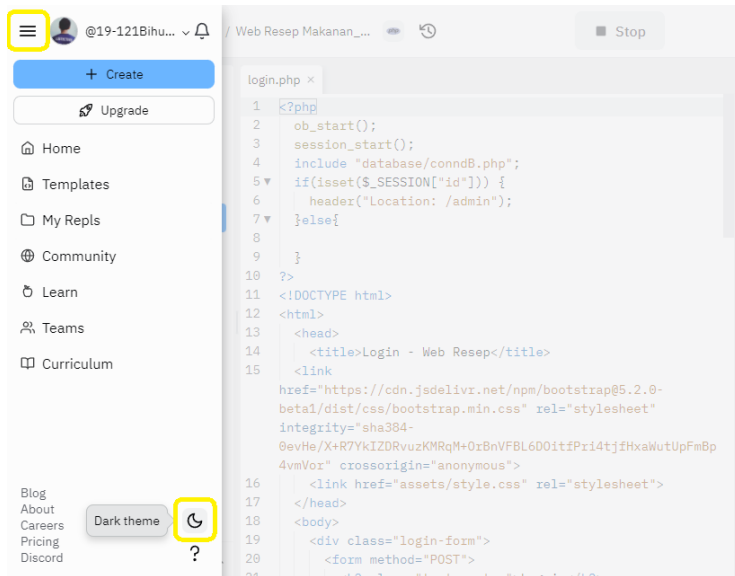
6.3.2 Pembuatan Interface *Login Admin*

Setelah selesai membuat modul koneksi database, selanjutnya kita akan membuat halaman login. Halaman ini adalah halaman yang dibuat dengan fungsi untuk melakukan proses autentikasi admin, sehingga dengan adanya halaman ini kita dapat menyeleksi siapa saja yang berhak mengakses sistem (terutama sebagai admin website).

* Saran penulis :

Karena pada section pembuatan *interface* ini dan selanjutnya akan mengimplementasikan banyak kode program, alangkah baiknya pembaca untuk mengubah tema Replit terlebih dahulu ke *dark theme* atau mode gelap, hal ini guna mengurangi kelelahan mata akibat warna latar yang terlalu terang.

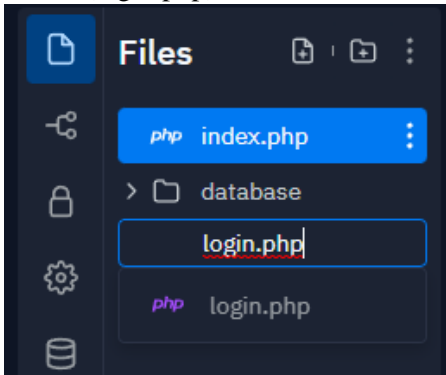
Untuk mengubahnya, silakan buka menu sidebar replit, kemudian klik logo atau gambar bulan, seperti pada gambar di bawah ini.



Gambar 6. 16 Mengubah Tema Replit Menjadi Mode Gelap

Berikut adalah langkah pembuatan *interface login admin* :

1. Buat terlebih dahulu berkas baru untuk *interface login admin* (login.php).



Gambar 6. 17 Membuat Berkas login.php

2. Masukkan kode berikut pada berkas login.php.

login.php
<pre><?php ob_start(); session_start(); include "database/conndB.php"; if(isset(\$_SESSION["id"])) { header("Location: /admin"); }else{ } ?> <!DOCTYPE html> <html></pre>

```

<head>
  <title>Login - Web Resep</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPr
i4tjfhHxaWutUpFmBp4vmVor"
crossorigin="anonymous">
  <link href="assets/style.css"
rel="stylesheet">
</head>
<body>
  <div class="login-form">
    <form method="POST">
      <h2 class="text-center">Log
in</h2>
      <div class="form-group">
        <input name="username"
type="text" class="form-control"
placeholder="Username"
required="required">
      </div>
      <div class="form-group">
        <input name="password"
type="password" class="form-control"
placeholder="Password"
required="required">
      </div>
      <div class="form-group">
        <button type="submit"
class="btn btn-primary btn-block">Log
in</button>
      </div>
    </form>
  </div>
</body>
</html>

```

```

        if($_SERVER["REQUEST_METHOD"] ==
"POST"){
            $username = $_POST["username"];
            $password = $_POST["password"];
            $cek_kredensial = $conndB-
>query("SELECT COUNT(*) as kredensial
FROM ADMIN WHERE USERNAME_ADMIN =
'$username' AND PASSWORD_ADMIN =
'$password'");
            $row = $cek_kredensial-
>fetchArray();
            $count = $row['kredensial'];
            if( $count > 0){

                $get_id = $conndB-
>query("SELECT ID_ADMIN FROM ADMIN WHERE
USERNAME_ADMIN = '$username' AND
PASSWORD_ADMIN = '$password'");
                $id = $get_id->fetchArray();
                $_SESSION["id"] = $id[0];
                header("Location: /admin",
true, 301);
                echo "sukses";
            } else {
                $conndB->query("insert or
ignore into ADMIN
values('adm01','bihubbil','bihubbil','nim
19-121')");
                echo '<h5 class="text-white
badge bg-primary" >gagal, username atau
password salah</h5>';
            }
        }
    ?>
</div>
</body>
</html>

```

Tabel 6. 17 Kode Untuk Berkas login.php

Kode pada berkas “login.php” diatas berfungsi untuk:

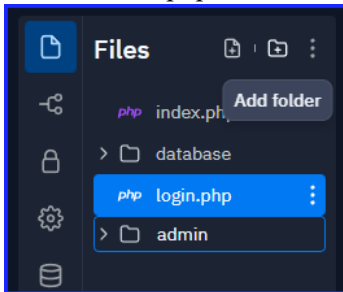
- o **Membuat fungsi cek login** : Berguna untuk mengarahkan langsung pengguna ke halaman *dashboard admin*, yang sebelumnya telah login, dan sesinya belum kadaluarsa.
- o **Membuat *post form*** : Berguna untuk mengirimkan data *inputan* kredensial pengguna.
- o **Membuat fungsi cek kredensial** : Berguna untuk menerima data dari *post form*, dan mengeceknya apakah data yang *diinputkan valid* atau tidak.
- o **Membuat fungsi *insert kredensial admin*** : Karena disini kita tidak menyediakan form pendaftaran admin, maka kita akan harus membuat suatu fungsi yang dapat menambahkan kredensial default. Fungsi ini akan berjalan ketika fungsi cek kredensial gagal melakukan autentikasi karena tidak adanya data admin pada *database*.

6.3.3 Pembuatan *Index Interface Admin*

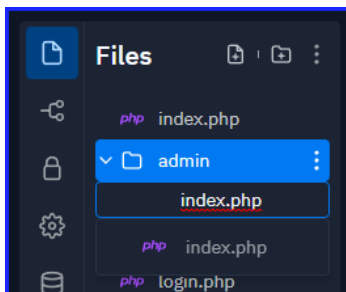
Setelah membuat *interface login*, selanjutnya kita akan membuat suatu interface untuk mengindeks seluruh berkas interface, yang nantinya akan dapat diakses oleh admin.

Berikut adalah langkah pembuatan *index interface admin* :

1. Buat direktori baru bernama “admin”, kemudian buat berkas “index.php” di dalamnya.



Gambar 6. 18 Membuat Direktori admin



Gambar 6. 19 Membuat Berkas admin/index.php

2. Masukkan kode berikut pada berkas index.php.

admin/index.php

```
<?php
    ob_start();
    session_start();
    $page = $_GET["page"];
    include "../database/conndB.php";
    if(isset($_SESSION["id"])){
        define("INDEX", true);
    }else{
        header("Location:login.php");
    }

    $content = array("dashboard",
    "kategori", "tambah_kategori",
    "edit_kategori", "hapus_kategori",
    "resep", "tambah_resep", "edit_resep",
    "hapus_resep", "permintaan",
    "tambahkan_keresep",
    "permintaan_duplikat");

    switch ($page) {
        case 'kategori':
            $title = "Data Kategori";
            break;
        case 'tambah_kategori':
            $title = "Tambah Kategori";
            break;
        case 'edit_kategori':
            $title = "Sunting Kategori";
            break;
        case 'resep':
            $title = "Data Resep";
            break;
        case 'tambah_resep':
```

```

        $title = "Tambah Resep";
        break;
    case 'edit_resep':
        $title = "Sunting Resep";
        break;
    case 'permintaan':
        $title = "Data Permintaan";
        break;
    case 'tambahkan_keresepe':
        $title = "Tambah Ke Resep";
        break;
    default:
        $title = "Dashboard";
        break;
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <link
href="https://cdn.jsdelivrivr.net/npm/bootst
rap@5.2.0-
beta1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPr
i4tjfhXaWutUpFmBp4vmVor"
crossorigin="anonymous">
    <link href="../../assets/style.css"
rel="stylesheet">
    <title>
    <?php
        echo $title;

```

```

    ?>
  </title>
</head>
<body>
  <div class="row">
    <div class="col-md-12 navbar-
dashboard">
      <h4>Web Resep Makanan</h4>
    </div>
  </div>
  <div class="row">
    <div class="col-md-3 sidebar">
      <div class="menu">
        <ul class="">
          <li>
            <a class="<?php
if($title == "Dashboard"){echo "active";}
?>" href="?page=dashboard">DASHBOARD</a>
          </li>

          <li>
            <a class="<?php
if($title == "Data Kategori" OR $title ==
"Tambah Kategori" OR $title == "Sunting
Kategori"){echo "active";} ?>"
href="?page=kategori">DATA KATEGORI</a>
          </li>

          <li>
            <a class="<?php
if($title == "Data Resep" OR $title ==
"Tambah Resep" OR $title == "Sunting
Resep"){echo "active";} ?>"
href="?page=resep">DATA RESEP</a>
          </li>

          <li>
            <a class="<?php
if($title == "Data Permintaan"){echo

```

```

"active";} ?>"
href="?page=permintaan">DATA
PERMINTAAN</a>

        </li>
        <li>
            <a
href="/logout.php">LOGOUT</a>
        </li>
    </ul>
</div>
</div>
<div class="col-md-9 contents">
    <?php
        if(!defined("INDEX")) die("");
        if(isset($_GET["page"])){
            $pages = $_GET["page"];
        } else{
            $pages = "dashboard";
        }

        foreach ($content as $konten) {
            if($pages == $konten){
                if($page == "kategori" OR
$page == "resep" OR $page ==
"permintaan") {?>
                    <form method="POST"
action="">

                        <?php
                            if($page ==
"permintaan"){

                                if(!empty($_POST['search'])) {

                                    $_SESSION['search'] = $_POST['search'];
                                }

                                if(!empty($_SESSION['search'])) {

```

```

        $_POST['search']
    = $_SESSION['search'];
        }

        }else{

unset($_SESSION['search']);
        }

        ?>
        <div class="col-md-4
offset-md-7 d-flex justify-content-end">
            <input class="form-
control me-2" type="text" name="search"
value="<?php echo $_POST['search']?>"
placeholder="Search">
            <button class="btn
btn-dark" type="submit">Search</button>
        </div>
    </form>
    <?php
    }

    include
"content/$konten.php";
    break;
    }
    }
    ?>
</div>
</div>
<div class="row">
    <footer class="col-md-12 footer">
        <p>Copyright &copy;
190411100121</p>
    </footer>
</div>
</body>

```

```
</html>
```

Tabel 6. 18 Kode Untuk Berkas *admin/index.php*

Kode pada berkas “*index.php*” diatas berfungsi untuk:

- o **Membuat Fungsi cek login** : Hampir sama dengan yang ada pada “*login.php*”, namun disini,, jika pengguna telah login, maka *variable* index akan didefinisikan dengan *value* true, pendefinisian tersebut nantinya berguna untuk memastikan pengguna tidak mengakses berkas *interface admin* lainnya secara langsung, jadi harus melalui *interface* index terlebih dahulu. Dan jika pengguna belum login, maka akan diarahkan ke *interface login admin*.
- o **Membuat daftar indeks *interface admin* dan pengondisian judul halaman** : Berguna untuk menginformasikan kepada sistem tentang *interface* apa saja yang akan terindeks, kemudian juga mengondisikan judul halaman sesuai indeks *interface* yang sedang diakses.
- o **Membuat form pencarian** : Berfungsi untuk melakukan pencarian data yang terdapat pada *interface* data kategori resep, resep, dan permintaan resep.

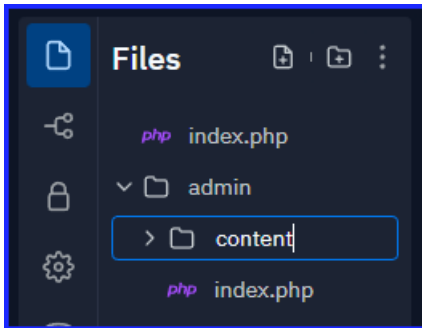
- o **Membuat *sidebar menu*** : Berfungsi untuk membuat navigasi yang dapat mengarah ke *interface admin* lainnya dan kemudian ditampilkan di *interface* ini.
- o **Menampilkan isi konten *interface admin* lainnya** : Berguna untuk menampilkan konten dari *interface admin* lainnya yang terdapat pada daftar indeks di *interface* ini.

6.3.4 Pembuatan *Interface Dashboard Admin*

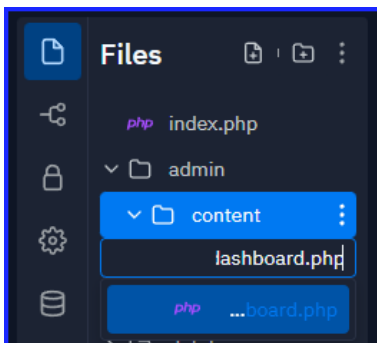
Interface ini merupakan interface utama yang akan ditampilkan jika pengguna berhasil login sebagai admin. Pada interface ini terdapat nama admin, jumlah data resep dan data permintaan resep.

Berikut adalah langkah pembuatan *interface dashboard admin*:

1. Buat direktori baru bernama “content” didalam direktori “admin”, kemudian buat berkas “dashboard.php” di dalamnya.



Gambar 6. 20 Membuat Direktori admin/content



Gambar 6. 21 Membuat Berkas admin/content/dashboard.php

2. Masukkan kode berikut pada berkas dashboard.php.

admin/content/dashboard.php

```
<?php
    if(!defined("INDEX")) die("");
    $get_data = $conndB-
    >query("SELECT * FROM ADMIN WHERE
    ID_ADMIN = '". $_SESSION["id"]."'
    ");
```

```

    $get_resep = $conndB-
    >query("SELECT COUNT(*) as
    jumlahresep FROM RESEP ");
    $get_permintaan = $conndB-
    >query("SELECT COUNT(*) as
    jumlahpermintaan FROM
    PERMINTAANRESEP ");
    $data = $get_data->fetchArray();
    $jumlah_resep = $get_resep-
    >fetchArray();
    $jumlah_permintaan =
    $get_permintaan->fetchArray();
    ?>
<div>
    <h3>Selamat datang, <?php echo
    $data["NAMA_ADMIN"]?></h3>
    <br>
    <div class="row">
    <div class="col-sm-6">
    <div class="card bg-info">
    <h5 class="card-header">Data
    Resep</h5>
    <div class="card-body">
    <h5 class="card-title">Jumlah
    Resep Saat Ini</h5>
    <p class="card-text"><?php
    echo
    $jumlah_resep["jumlahresep"]?></p>
    </div>
    </div>
    </div>
    <div class="col-sm-6">
    <div class="card bg-warning">
    <h5 class="card-header">Data
    Permintaan</h5>
    <div class="card-body">
    <h5 class="card-title">Jumlah
    Permintaan Saat Ini</h5>

```

```

        <p class="card-text"><?php
        echo
        $jumlah_permintaan["jumlahpermintaan"]?></p>
        </div>
        </div>
        </div>
        </div>
        </div>
    </div>

```

Tabel 6. 19 Kode Untuk Berkas *admin/content/dashboard.php*

Kode pada berkas “*dashboard.php*” diatas berfungsi untuk:

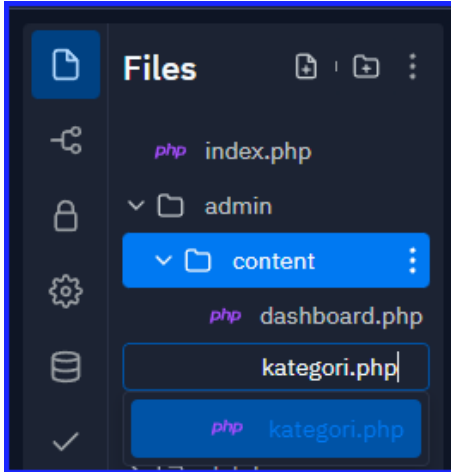
- o **Membuat Fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface dashboard* ini diakses melalui *index interface admin*, bukan langsung melalui berkas *dashboard.php*.
- o **Mengambil dan menampilkan data dari *database*** : Data nama admin, beserta jumlah resep dan permintaan resep akan diambil dari database, kemudian ditampilkan pada *interface* ini.

6.3.5 Pembuatan *Interface* Daftar Kategori Resep

Interface daftar kategori resep merupakan *interface* yang berisi daftar kategori resep apa saja yang ada pada *database website*.

Berikut adalah langkah pembuatan *interface* daftar kategori resep:

1. Buat berkas baru bernama “kategori.php” di dalam direktori “content”.



Gambar 6. 22 Membuat Berkas admin/content/kategori.php

2. Masukkan kode berikut pada berkas kategori.php.

admin/content/kategori.php
<pre><?php if(!defined("INDEX")) die(""); \$get_data = \$conndB->query("SELECT * FROM KATEGORIRESSEP"); ?> <div class="col-md-11"> <div class="content-head"> <h1>Data Kategori</h1></pre>

```

</div>
<div class="content-button">
  <a href="?page=tambah_kategori"
class="btn btn-primary">Tambah</a>
  <br>
  <p>
    <?php
      echo $_SESSION["pesan"];
      unset($_SESSION["pesan"]);
    ?>
  </p>
</div>
<div class="content-table">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>No</th>
        <th>Nama Kategori</th>
        <th>Aksi</th>
      </tr>
    </thead>
    <tbody>
      <?php
        $i = 0;
        while($data = $get_data-
>fetchArray()) {
          $i++;
          ?>
          <tr>
            <td><?php echo $i;?></td>
            <td><?php echo
$data["NAMA_KATEGORI"];?></td>
            <td><?php
              echo '<a class="btn btn-
warning"
href="?page=edit_kategori&id_kategori='.$
data["ID_KATEGORI"].' ">Sunting</a>

```

```

                                <a class="btn btn-
danger"
href="?page=hapus_kategori&id_kategori='
$data["ID_KATEGORI"].' ">Hapus</a>';
                                ?>
                                </td>
                                </tr>
                                <?php
                                };
                                ?>
                                </tbody>
                                </table>
                                </div>
                                </div>

```

Tabel 6. 20 Kode Untuk Berkas *admin/content/kategori.php*

Kode pada berkas “kategori.php” diatas berfungsi untuk:

- o **Membuat Fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface* daftar kategori resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas kategori.php.
- o **Mengambil dan menampilkan data dari *database*** : Data nama kategori akan diambil dari database, kemudian ditampilkan pada *interface* ini.
- o **Tombol tambah, sunting, dan hapus** : Berfungsi untuk dapat melakukan penambahan kategori resep baru, serta

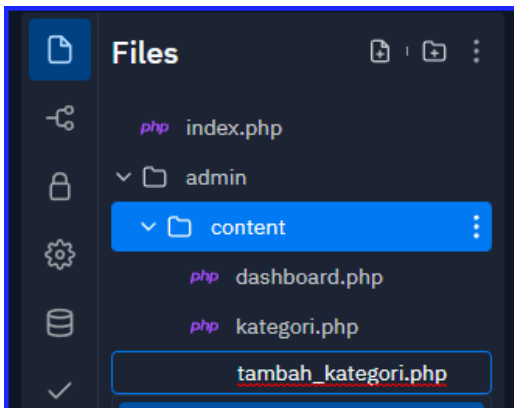
menyunting dan menghapus kategori resep yang sudah ada.

6.3.6 Pembuatan *Interface* Tambah Kategori Resep

Interface tambah kategori resep merupakan *interface* yang berisi form untuk melakukan penambahan data kategori resep pada *database website*.

Berikut adalah langkah pembuatan *interface* tambah kategori resep :

1. Buat berkas baru bernama “tambah_kategori.php” di dalam direktori “content”.



Gambar 6. 23 Membuat Berkas admin/content/tambah_kategori.php

2. Masukkan kode berikut pada berkas tambah_kategori.php.

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

admin/content/tambah_kategori.php

```
<?php
    if(!defined("INDEX")) die("");
?>
<div class="col-md-11">
    <div class="content-head">
        <h1>Tambah Kategori</h1>
    </div>
    <br>
    <div class="container content-form">
        <p>
            <?php
                echo $_SESSION["pesan"];
                unset($_SESSION["pesan"]);
            ?>
        </p>
        <form action="?page=tambah_kategori"
method="POST">
            <div class="form-group">
                <label for="nama_kategori">Nama
Kategori</label>
                <input class="form-control"
id="nama_kategori" name="nama_kategori"
type="text" placeholder="Masukkan nama
Kategori..." required>
            </div>
            <div class="form-group">
                <input class="btn btn-success"
id="submit" type="submit" value="Tambah">
                <a class="btn btn-danger"
href="?page=kategori">Batal</a>
            </div>
        </form>
    </div>
</div>
<?php
```



```

        if($_SERVER["REQUEST_METHOD"] == "POST"
        AND !empty($_POST["nama_kategori"])){
            $get_data = $conndB->query("SELECT
            COUNT(*) as jumlahkategori FROM
            KATEGORIRESEP WHERE NAMA_KATEGORI =
            '$_POST["nama_kategori"]'");
            $cek_data = $get_data->fetchArray();
            if($cek_data['jumlahkategori'] == 0){
                $prefix = "kat";
                $get_data = $conndB->query("SELECT
                * FROM KATEGORIRESEP ORDER BY ID_KATEGORI
                DESC");
                $data = $get_data->fetchArray();
                if($data[0] == NULL){
                    $prefix = "kat01";
                }else{
                    $get_new_int =
                    intval(substr($data["ID_KATEGORI"],
                    3)+1);

                    if(intval(substr($data["ID_KATEGORI"], 3,
                    1)) == 0 AND
                    intval(substr($data["ID_KATEGORI"], 4,
                    1)) != 9){
                        $prefix =
                        $prefix."0"."$get_new_int";
                    }else{
                        $prefix =
                        $prefix."$get_new_int";
                    }
                }
                $insert_data = $conndB-
                >query("INSERT INTO KATEGORIRESEP
                VALUES('$_prefix.'',
                '$_POST["nama_kategori"]'");

                if($insert_data){

```

```

        $_SESSION["pesan"] = "<span
style='color:blue;'>Berhasil :</span>
Kategori (".$_POST["nama_kategori"].")
berhasil ditambahkan";
        echo "<meta http-
equiv='refresh'
content='0;url=?page=kategori'>";
    }
    }else {
        $_SESSION["pesan"] = "<span
style='color:red;'>Gagal :</span>
Kategori (".$_POST["nama_kategori"].")
sudah terdapat dalam database";
        echo "<meta http-equiv='refresh'
content='0;url=?page=tambah_kategori'>";
    }
}
?>

```

Tabel 6. 21 Kode Untuk Berkas *admin/content/tambah_kategori.php*

Kode pada berkas “*tambah_kategori.php*” diatas berfungsi untuk:

- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface* tambah kategori resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas *tambah_kategori.php*.
- o **Membuat *post form*** : Berguna untuk mengirimkan data kategori resep yang ingin ditambahkan oleh *admin*.

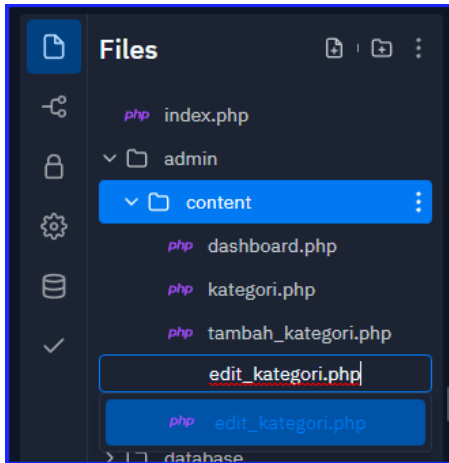
- o **Membuat fungsi cek kategori** : Berguna untuk menerima data dari *post form*, dan mengeceknya apakah data kategori resep yang diinputkan telah terdapat dalam *database* atau tidak.
- o **Membuat fungsi unik id kategori** : Berfungsi untuk membuat *id* kategori resep yang unik dan berbeda dengan table lainnya pada *database* (menambahkan *prefix* pada *id* kategori resep)
- o **Membuat fungsi insert data kategori** : Berfungsi untuk menambahkan data kategori resep yang telah diterima dari *post form* beserta *id* kategori resep unik yang telah dibuat sebelumnya, ke dalam *database website*.

6.3.7 Pembuatan *Interface* Sunting Kategori Resep

Interface sunting kategori resep merupakan *interface* yang berisi form untuk melakukan penyuntingan data kategori resep pada *database website*.

Berikut adalah langkah pembuatan *interface* sunting kategori resep :

1. Buat berkas baru bernama “edit_kategori.php” di dalam direktori “content”.



Gambar 6. 24 Membuat Berkas admin/content/edit_kategori.php

2. Masukkan kode berikut pada berkas edit_kategori.php.

admin/content/edit_kategori.php
<pre><?php if(!defined("INDEX")) die(""); if(!empty(\$_GET['id_kategori'])){ \$get_data = \$conndB->query("SELECT * FROM KATEGORIRESEP WHERE ID_KATEGORI = '".\$_GET['id_kategori']."' "); \$data = \$get_data->fetchArray(); }else{ header("Location:?page=kategori"); } ?> <div class="col-md-11"></pre>

```

<div class="content-head">
  <h1>Sunting Kategori</h1>
</div>
<br>
<div class="container content-form">
  <p>
    <?php
      echo $_SESSION["pesan"];
      unset($_SESSION["pesan"]);
    ?>
  </p>
  <form action="?page=edit_kategori"
method="POST">
    <div class="form-group">
      <label for="nama_kategori">Nama
Kategori</label>
      <input class="form-control"
id="nama_kategori" name="nama_kategori"
type="text" placeholder="Masukkan nama
Kategori..." value="<?php echo
$data['NAMA_KATEGORI']?>" required>
    </div>

    <div class="form-group">
      <input class="form-control"
id="id_kategori" name="id_kategori"
type="hidden" value="<?php echo
$data['ID_KATEGORI']?>" required>
      <input class="btn btn-success"
id="submit" type="submit" value="Simpan">
      <a class="btn btn-danger"
href="?page=kategori">Batal</a>
    </div>
  </form>
</div>
</div>
<?php

```

```

        if($ _SERVER["REQUEST_METHOD"] == "POST"
        AND !empty($_POST["nama_kategori"])){
            $get_data = $conndB->query("SELECT
            COUNT(*) as jumlahkategori FROM
            KATEGORIRESSEP WHERE NAMA_KATEGORI =
            '". $_POST["nama_kategori"]." '");
            $cek_data = $get_data->fetchArray();

            if($cek_data['jumlahkategori'] == 0){
                $update_data = $conndB-
                >query("UPDATE KATEGORIRESSEP SET
                NAMA_KATEGORI =
                '". $_POST["nama_kategori"]." ' WHERE
                ID_KATEGORI =
                '". $_POST['id_kategori']."'");

                if($update_data){
                    $_SESSION["pesan"] = "<span
                    style='color:blue;'>Berhasil :</span>
                    Kategori ('. $_POST["id_kategori"]."')
                    berhasil disunting";
                    echo "<meta http-equiv='refresh'
                    content='0;url=?page=kategori'>";
                }else {
                    $_SESSION["pesan"] = "<span
                    style='color:red;'>Gagal :</span>
                    Kategori ('. $_POST["id_kategori"]."')
                    gagal disunting";
                    echo "<meta http-equiv='refresh'
                    content='0;url=?page=kategori'>";
                }
            }else {
                $_SESSION["pesan"] = "<span
                style='color:red;'>Gagal :</span>
                Kategori ('. $_POST["nama_kategori"]."')
                sudah terdapat dalam database";
                echo "<meta http-equiv='refresh'
                content='0;url=?page=kategori'>";
            }
        }
    }

```

```

    }

}
?>

```

Tabel 6. 22 Kode Untuk Berkas `admin/content/edit_kategori.php`

Kode pada berkas “`edit_kategori.php`” diatas berfungsi untuk:

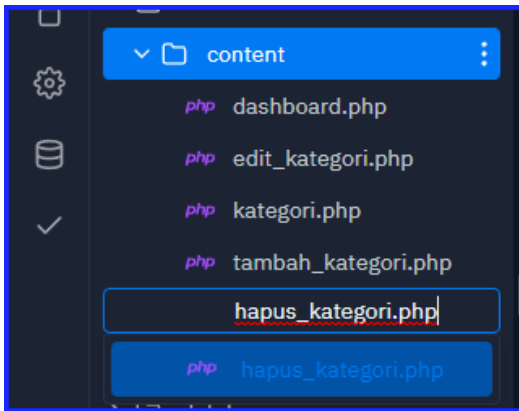
- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface* sunting kategori resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas `edit_kategori.php`.
- o **Mengambil data dari *database*** : Berfungsi untuk mengambil data kategori yang ingin disunting oleh admin, dari *database website*.
- o **Membuat *post form*** : Berguna untuk menampilkan dan mengirimkan data kategori yang ingin disunting oleh *admin*.
- o **Membuat fungsi *update kategori*** : Berguna untuk menerima data kategori dari *post form*, kemudian memperbaruinya pada database sesuai dengan data yang dikirimkan tersebut,

6.3.8 Pembuatan Fungsi Hapus Kategori Resep

Fungsi hapus kategori resep merupakan fungsi yang berguna untuk melakukan penghapusan data kategori pada *database website*.

Berikut adalah langkah pembuatan fungsi hapus kategori resep :

1. Buat berkas baru bernama “hapus_kategori.php” di dalam direktori “content”.



Gambar 6. 25 Membuat Berkas admin/content/hapus_kategori.php

2. Masukkan kode berikut pada berkas `hapus_kategori.php`.

admin/content/hapus_kategori.php
<pre><?php if(!defined("INDEX")) die(""); if(!empty(\$_GET["id_kategori"])){ \$get_data = \$conndB->query("SELECT * FROM KATEGORIASEP WHERE ID_KATEGORI = '".\$_GET['id_kategori']."' "); \$data = \$get_data->fetchArray(); \$conndB->exec("PRAGMA foreign_keys=on"); \$delete = \$conndB->query("DELETE FROM KATEGORIASEP WHERE ID_KATEGORI = '".\$_GET['id_kategori']."' "); if(\$delete){ \$_SESSION["pesan"] = "Berhasil : Kategori ".\$_data["NAMA_KATEGORI"]."[".\$_GET["id_ka tegori"]."] berhasil dihapus"; echo "<meta http-equiv='refresh' content='0;url=?page=kategori'>"; } \$conndB->exec("PRAGMA foreign_keys=off"); } ?></pre>

Tabel 6. 23 Kode Untuk Berkas `admin/content/hapus_kategori.php`

Kode pada berkas “`hapus_kategori.php`” diatas berfungsi untuk:

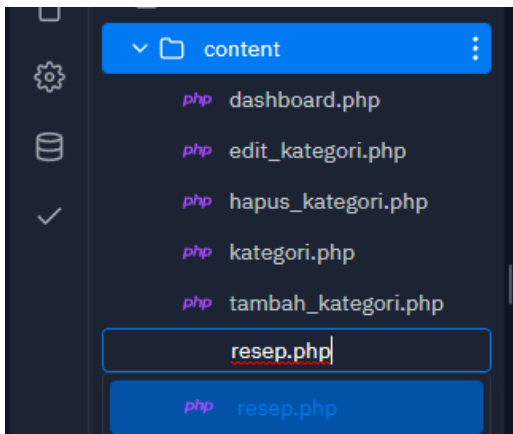
- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa fungsi hapus kategori resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas *hapus_kategori.php*.
- o **Membuat fungsi hapus kategori** : Berguna untuk menerima data *id* kategori dari *parameter url*, kemudian menghapus kategori dengan *id* tersebut pada *database*.

6.3.9 Pembuatan *Interface* Daftar Resep

Interface daftar resep merupakan *interface* yang berisi daftar resep apa saja yang ada pada *database website*.

Berikut adalah langkah pembuatan *interface* daftar resep:

1. Buat berkas baru bernama “resep.php” di dalam direktori “content”.



Gambar 6. 26 Membuat Berkas admin/content/resep.php

2. Masukkan kode berikut pada berkas resep.php.

```
admin/content/resep.php
<?php
    if(!defined("INDEX")) die("");
    $get_data = $conndB->query("SELECT *
FROM RESEP LEFT JOIN KATEGORI RESEP ON
KATEGORI RESEP.ID_KATEGORI =
RESEP.ID_KATEGORI WHERE NAMA RESEP LIKE
'%" . $_POST['search'] . "%' ORDER BY TANGGAL
DESC");

?>
<div class="col-md-11">
    <div class="content-head">
        <h1>Data RESEP</h1>
    </div>
    <div class="content-button">
```

```

<a href="?page=tambah_resep"
class="btn btn-primary">Tambah</a>
<br>
<p>
    <?php
        echo $_SESSION["pesan"];
        unset($_SESSION["pesan"]);
    ?>
</p>
</div>
<div class="content-table">
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>No</th>
                <th>Nama Resep</th>
                <th>Kategori</th>
                <th>Tanggal Ditambahkan</th>
                <th>Aksi</th>
            </tr>
        </thead>
        <tbody>
            <?php
                $i = 0;
                while($data = $get_data-
>fetchArray()) {
                    $i++
                    ?>
                    <tr>
                        <td><?php echo $i;?></td>
                        <td><?php echo
$data["NAMA_RESEP"];?></td>
                        <td>
                            <?php
                                if ($data["ID_KATEGORI"] ==
"" ) {
                                    echo "Tidak terkategori";
                                } else {

```

```

        echo
        $data["NAMA_KATEGORI"];
    }
    ?>
</td>
<td><?php echo
$data["TANGGAL"];?></td>

<td><?php
        echo '<a class="btn btn-
warning"
href="?page=edit_resep&id_resep='.$data["
ID_RESEP"].' ">Sunting</a>
        <a class="btn
btn-danger"
href="?page=hapus_resep&id_resep='.$data[
"ID_RESEP"].' ">Hapus</a>';
    ?>
</td>
</tr>
<?php
    };
    ?>
</tbody>
</table>
</div>
</div>

```

Tabel 6. 24 Kode Untuk Berkas `admin/content/resep.php`

Kode pada berkas “resep.php” diatas berfungsi untuk:

- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface* daftar resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas `resep.php`.

o **Mengambil dan menampilkan data dari *database* :**

Data nama, kategori, dan tanggal resep ditambahkan akan diambil dari database, kemudian ditampilkan pada *interface* ini.

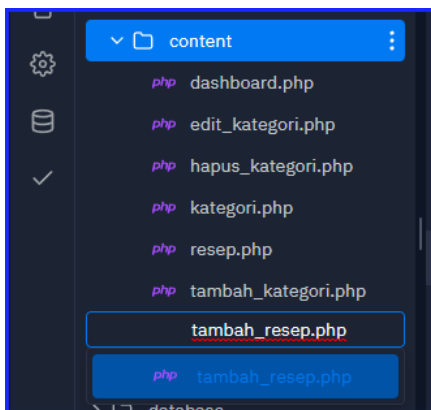
o **Tombol tambah, sunting, dan hapus :** Berfungsi untuk dapat melakukan penambahan resep baru, serta menyunting dan menghapus resep yang sudah ada.

6.3.10 Pembuatan *Interface* Tambah Resep

Interface tambah resep merupakan *interface* yang berisi form untuk melakukan penambahan data resep pada *database website*.

Berikut adalah langkah pembuatan *interface* tambah resep :

1. Buat berkas baru bernama “tambah_resep.php” di dalam direktori “content”.



Gambar 6. 27 Membuat Berkas admin/content/tambah_resep.php

2. Masukkan kode berikut pada berkas tambah_resep.php.

admin/content/tambah_resep.php
<pre> <?php if(!defined("INDEX")) die(""); ?> <div class="col-md-11"> <div class="content-head"> <h1>Tambah Resep</h1> </div>
 <div class="container content-form"> <p> <?php echo \$_SESSION["pesan"]; unset(\$_SESSION["pesan"]); ?> </p> <?php if(isset(\$_POST['id_permintaan'])) { </pre>

```

        $get_data = $conndB->query("SELECT
* FROM PERMINTAANRESEP LEFT JOIN
KATEGORIRESEP ON
PERMINTAANRESEP.ID_KATEGORI =
KATEGORIRESEP.ID_KATEGORI WHERE
ID_PERMINTAAN =
'".$_POST['id_permintaan']."' ");
        $data = $get_data->fetchArray();
        if(!empty($data['0'])){?>
            <div class="d-flex justify-
content-end">
                <h5 class="badge bg-
primary"><?php echo 'Ditambahkan dari
permintaan';?></h5>
            </div>
            <?php
            }
        }?>

        <form action="?page=tambah_resep"
method="POST" enctype="multipart/form-
data">
            <div class="form-group">
                <label for="nama_resep">Nama
Resep</label>
                <input class="form-control"
id="nama_resep" name="nama_resep"
type="text" value="<?php echo
$data['NAMA_PERMINTAAN'];?>"
placeholder="Masukkan nama resep.."
required>
            </div>
            <div class="form-group">
                <label for="foto_resep">Foto
Resep</label>
                <input class="form-control"
id="foto_resep" name="foto_resep"
type="file" accept="image/*" required>

```



```

        </div>

        <div class="form-group">
            <label
for="deskripsi_resep">Deskripsi Resep
</label>
            <textarea class="form-control"
id="deskripsi_resep"
name="deskripsi_resep"
placeholder="Deskripsi.." required><?php
echo
$data['CATATAN_PERMINTAAN'];?></textarea>
        </div>

        <div class="form-group">
            <label
for="kategori_resep">Kategori
Resep</label>
            <select class="form-control"
id="kategori_resep" name="kategori_resep"
required>
                <option value="<?php echo
$data['ID_KATEGORI'] ?>"><?php
if(!empty($data['NAMA_KATEGORI'])) {echo
$data['NAMA_KATEGORI'];} else {echo
"Silakan Pilih Kategori Resep";}
?></option>
                <?php
                    $get_alldata = $conndB-
>query("SELECT * FROM KATEGORIRESSEP");
                    while($alldata =
                    $get_alldata->fetchArray()) {
                        if($data['ID_KATEGORI']
!= $alldata['ID_KATEGORI']) {
                            echo "<option
value='".$alldata['ID_KATEGORI']."'>".$al
ldata['NAMA_KATEGORI']."'</option>";
                        }
                    }
            </select>
        </div>
    </div>

```

```

    }

    ?>
  </select>
</div>
<div class="form-group">
  <label for="bahan_resep">Bahan
  (untuk setiap bahan, pisahkan dengan
  baris baru)</label>
  <textarea class="form-control"
  id="bahan_resep" name="bahan_resep"
  placeholder="Bahan.."
  required></textarea>
</div>
<div class="form-group">
  <label for="cara_memasak">Cara
  Memasak (untuk setiap langkah, pisahkan
  dengan baris baru)</label>
  <textarea rows="5" class="form-
  control" id="cara_memasak"
  name="cara_memasak" placeholder="Cara
  memasak.." required></textarea>
</div>
<div class="form-group">
  <input id="id_permintaan"
  name="id_permintaan" type="hidden"
  value="<?php echo
  $data['ID_PERMINTAAN'];?>">
  <input id="nama_permintaan"
  name="nama_permintaan" type="hidden"
  value="<?php echo
  $data['NAMA_PERMINTAAN'];?>">
  <input class="btn btn-success"
  id="submit" type="submit" value="Tambah">
  <a class="btn btn-danger"
  href="?page=resep">Batal</a>
</div>

```

```

        </form>
    </div>
</div>
<?php
    if(!defined("INDEX")) die("");
    if($_SERVER["REQUEST_METHOD"] == "POST"
    AND !empty($_POST["nama_resep"]) AND
    !empty($_FILES["foto_resep"])) ){
        #buat uniq id
        $prefix = "res";
        $get_data = $conndb->query("SELECT *
    FROM RESEP ORDER BY TANGGAL DESC");
        $data = $get_data->fetchArray();
        if($data[0] == NULL){
            $prefix = "res1";
        }else{
            $get_new_int =
    intval(substr($data["ID_RESEP"], 3)+1);
            $prefix = $prefix."$get_new_int";
        }
        #mendapatkan data untuk insert ke
    table resep

        $tipe_file =
    $_FILES["foto_resep"]["type"];
        if($tipe_file == "image/jpeg" OR
    $tipe_file == "image/jpg" OR $tipe_file
    == "image/png" OR $tipe_file ==
    "image/webp"){
            $nama_foto =
    $_FILES["foto_resep"]["name"];
            $loc_foto =
    $_FILES["foto_resep"]["tmp_name"];
            $ext_foto = pathinfo($nama_foto,
    PATHINFO_EXTENSION);

            $nama_ext = $prefix.".".$ext_foto;

```

```

        $new_loc =
$_SERVER['DOCUMENT_ROOT']."/assets/images
/";

        move_uploaded_file($loc_foto,
$new_loc.$nama_ext);

        $insert_data = $conndB-
>query("INSERT INTO RESEP
VALUES('".$_prefix."',
'".$_POST["id_permintaan"]."',
'".$_POST["kategori_resep"]."',
'".$_SESSION["id"]."',
'".$_POST["nama_resep"]."',
'".$_POST["deskripsi_resep"]."',
'".$_POST["bahan_resep"]."',
'".$_POST["cara_memasak"]."',
'".$_nama_ext."', strftime('%Y-%m-%d %H-
%M-%S', 'now') )");
    }

    if($insert_data){

if(!empty($_POST["id_permintaan"])){
        $update_data = $conndB-
>query("UPDATE PERMINTAANRESEP SET
ID_STATUS = 'sts02' WHERE ID_PERMINTAAN =
'".$_POST["id_permintaan"]."' ");

        if($update_data){
            $_SESSION["pesan"] = "<span
style='color:blue;'>Berhasil </span>
Permintaan '".$_POST["nama_permintaan"]."
('".$_POST["id_permintaan"].") berhasil
ditambahkan ke resep";
            $_SESSION['filter'] = 'sts02';
            echo "<meta http-
equiv='refresh'
content='0;url=?page=permintaan'>";

```

```

    }
    }else{
        $_SESSION["pesan"] = "<span
style='color:blue;'>Berhasil :</span>
Resep
".$_POST['nama_resep'].("'.$prefix.'")
berhasil ditambahkan";
        echo "<meta http-equiv='refresh'
content='0;url=?page=resep'>";
    }

    }

}

?>

```

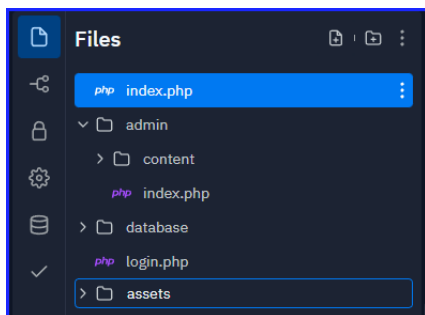
Tabel 6. 25 Kode Untuk Berkas `admin/content/tambah_resep.php`

Kode pada berkas “`tambah_resep.php`” diatas berfungsi untuk:

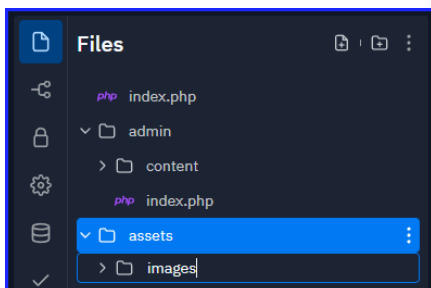
- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface* tambah resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas `tambah_resep.php`.
- o **Membuat *post form*** : Berguna untuk mengirimkan data resep yang ingin ditambahkan oleh *admin*.
- o **Membuat fungsi cek berkas gambar** : Berguna untuk menerima data dari *post form*, dan mengeceknya apakah berkas yang dikirimkan *admin* merupakan berkas

gambar atau bukan. Jika berkas merupakan gambar, maka berkas gambar akan disimpan pada direktori “assets/images”.

- o **Membuat fungsi unik *id* resep** : Berfungsi untuk membuat *id* resep yang unik dan berbeda dengan table lainnya pada *database* (menambahkan *prefix* pada *id* resep)
 - o **Membuat fungsi *insert data* resep** : Berfungsi untuk menambahkan data resep yang telah diterima dari *post form* beserta *id* resep unik yang telah dibuat sebelumnya, ke dalam *database website*.
 - o **Membuat fungsi *update status* permintaan resep** : Berfungsi untuk mengubah status dari permintaan resep yang sebelumnya “belum ditambahkan” menjadi “telah ditambahkan”. Fungsi ini hanya akan berjalan jika resep yang akan ditambahkan, merupakan resep yang berasal dari permintaan pengunjung.
3. Buat direktori baru bernama “assets”, kemudian buat kembali direktori “images” di dalamnya.



Gambar 6. 28 Membuat Direktori assets



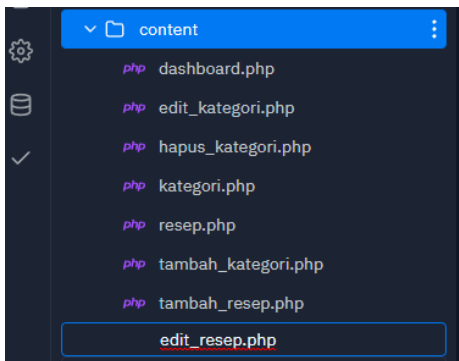
Gambar 6. 29 Membuat Direktori image/assets

6.3.11 Pembuatan *Interface* Sunting Resep

Interface sunting resep merupakan *interface* yang berisi form untuk melakukan penyuntingan data resep pada *database website*.

Berikut adalah langkah pembuatan *interface* sunting kategori resep :

1. Buat berkas baru bernama “edit_resep.php” di dalam direktori “content”.



Gambar 6. 30 Membuat Berkas admin/content/edit_resep.php

2. Masukkan kode berikut pada berkas edit_resep.php.

admin/content/edit_resep.php

```
<?php
if(!defined("INDEX")) die("");
if(!empty($_GET['id_resep'])){
    $get_data = $conndB->query("SELECT *
FROM RESEP LEFT JOIN KATEGORI RESEP ON
```



```

KATEGORI RESEP.ID_KATEGORI =
RESEP.ID_KATEGORI WHERE ID_RESEP =
'".$_GET['id_resep']."' ";
    $data = $get_data->fetchArray();
} else {
    header("Location: ?page=resep");
}
?>
<div class="col-md-11">
    <div class="content-head">
        <h1>Sunting Resep</h1>
    </div>
    <br>
    <div class="container content-form">
        <p>
            <?php
                echo $_SESSION["pesan"];
                unset($_SESSION["pesan"]);
            ?>
        </p>
        <?php

if(!empty($data["ID_PERMINTAAN"])) { ?>
    <div class="d-flex justify-
content-end">
        <h5 class="badge bg-
primary"><?php echo "id permintaan:
".$data["ID_PERMINTAAN"]; ?></h5>
    </div>
    <?php
    }
    ?>
    <form action="?page=edit_resep"
method="POST" enctype="multipart/form-
data">
        <div class="form-group">
            <label for="nama_resep">Nama
Resep</label>

```

```

        <input class="form-control"
id="nama_resep" name="nama_resep"
type="text" placeholder="Masukkan nama
resep.." value="<?php echo
$data["NAMA_RESEP"] ?>" required>
    </div>
    <div class="form-group">
        <label>Foto Resep</label>
        <div style="height:310px; width:
fit-content; padding-top:5px; background-
color: grey;">
            
        </div>
        <label for="foto_resep" >
            <div class="d-flex btn btn-
primary">
                <span class="text-white mb-
0">Ubah Gambar</span>
            </div><br>
        </label>
        <input class="form-control"
id="foto_resep" name="foto_resep"
type="file" style="display:none">
    </div>
    <div class="form-group">
        <label
for="deskripsi_resep">Deskripsi
Resep</label>
        <textarea rows="4" class="form-
control" id="deskripsi_resep"
name="deskripsi_resep"
placeholder="Deskripsi.." required><?php
echo $data["DESKRIPSI"] ?></textarea>
    </div>
    <div class="form-group">

```

```

        <label
for="kategori_resep">Kategori
Resep</label>
        <select class="form-control"
id="kategori_resep" name="kategori_resep"
required>
            <option value="<?php echo
$data["ID_KATEGORI"] ?>"><?php echo
$data["NAMA_KATEGORI"] ?></option>
            <?php
                $get_alldata = $conndB-
>query("SELECT * FROM KATEGORIRESSEP");
                while($alldata =
                $get_alldata->fetchArray()){
                    if($data["ID_KATEGORI"]
!= $alldata["ID_KATEGORI"]){
                        echo "<option
value='".$alldata["ID_KATEGORI"]."'.>".$al
ldata['NAMA_KATEGORI']."</option>";
                    }
                }
            ?>
        </select>
    </div>
    <div class="form-group">
        <label for="bahan_resep">Bahan
(untuk setiap bahan, pisahkan dengan
baris baru)</label>
        <textarea rows="7" class="form-
control" id="bahan_resep"
name="bahan_resep" placeholder="Bahan.."
required><?php echo $data["BAHAN"]
?></textarea>
    </div>
    <div class="form-group">
        <label for="cara_memasak">Cara
Memasak (untuk setiap langkah, pisahkan
dengan baris baru)</label>

```

```

        <textarea rows="10"
class="form-control" id="cara_memasak"
name="cara_memasak" placeholder="Cara
memasak.." required><?php echo
$data["CARAMEMASAK"] ?></textarea>
    </div>
    <div class="form-group">
        <input class="form-control"
id="id_resep" name="id_resep"
type="hidden" value="<?php echo
$_GET['id_resep'] ?>" required>
        <input class="btn btn-success"
id="submit" type="submit" value="Simpan">
        <a class="btn btn-danger"
href="?page=resep">Batal</a>
    </div>
</form>
</div>
</div>
<?php
    if($_SERVER["REQUEST_METHOD"] == "POST"
AND !empty($_POST["nama_resep"])){
        #mendapatkan data untuk insert ke
table resep
        $tipe_file =
$_FILES["foto_resep"]["type"];
        if($tipe_file == "" OR $tipe_file ==
NULL){
            $update_data = $conndB-
>query("UPDATE RESEP SET ID_KATEGORI =
'".$_POST["kategori_resep"]."',
NAMA RESEP = '".$_POST["nama_resep"]."',
DESKRIPSI =
'".$_POST["deskripsi_resep"]."', BAHAN =
'".$_POST["bahan_resep"]."', CARAMEMASAK
= '".$_POST["cara_memasak"]."' WHERE
ID_RESEP = '".$_POST['id_resep']."'");
        } else{

```

```

        if($stipe_file == "image/jpeg" OR
        $stipe_file == "image/jpg" OR $stipe_file
        == "image/png"){
            $nama_foto =
            $_FILES["foto_resep"]["name"];
            $loc_foto =
            $_FILES["foto_resep"]["tmp_name"];
            $ext_foto = pathinfo($nama_foto,
            PATHINFO_EXTENSION);
            $get_oldname = $conndB-
            >query("SELECT * FROM RESEP WHERE
            ID_RESEP = '". $_POST['id_resep']."'");
            $oldname = $get_oldname-
            >fetchArray();
            $nama_ext =
            $oldname['ID_RESEP']. "." . $ext_foto;
            $new_loc =
            $_SERVER['DOCUMENT_ROOT'] . "/assets/images
            /";

            unlink($new_loc.$oldname['GAMBAR']);
            move_uploaded_file($loc_foto,
            $new_loc.$nama_ext);
            $update_data = $conndB-
            >query("UPDATE RESEP SET ID_KATEGORI =
            '". $_POST["kategori_resep"]."',
            NAMA_RESEP = '". $_POST["nama_resep"]."',
            DESKRIPSI =
            '". $_POST["deskripsi_resep"]."', BAHAN =
            '". $_POST["bahan_resep"]."', CARAMEMASAK
            = '". $_POST["cara_memasak"]."', GAMBAR =
            '". $nama_ext."' WHERE ID_RESEP =
            '". $_POST['id_resep']."'");
        }
    }
    if($update_data){
        $_SESSION["pesan"] = "<span
        style='color:blue;'>Berhasil :</span>

```

```

Resep ("$_POST["id_resep"].") berhasil
disunting";
    echo "<meta http-equiv='refresh'
content='0;url=?page=resep'>";
    }
}
?>

```

Tabel 6. 26 Kode Untuk Berkas *admin/content/edit_resep.php*

Kode pada berkas “*edit_resep.php*” diatas berfungsi untuk:

- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface* sunting resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas *edit_resep.php*.
- o **Membuat *post form*** : Berguna untuk menampilkan dan mengirimkan data resep yang ingin disunting oleh *admin*.
- o **Membuat fungsi cek berkas gambar** : Berguna untuk menerima data dari *post form*, dan mengeceknya apakah berkas yang dikirimkan *admin* merupakan berkas gambar atau bukan. Jika berkas merupakan gambar, maka berkas gambar sebelumnya akan dihapus, dan diganti dengan berkas gambar yang baru.

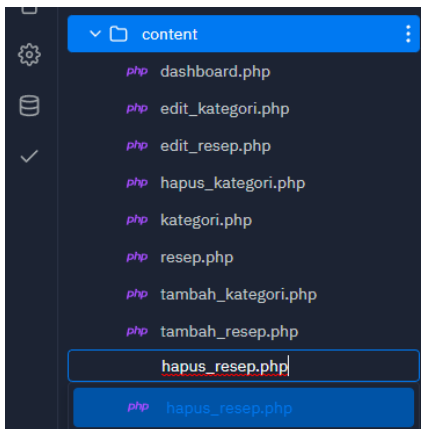
- o **Membuat fungsi *update* resep** : Berguna untuk menerima data resep dari *post form*, kemudian memperbaruinya pada database sesuai dengan data resep yang dikirimkan tersebut,

6.3.12 Pembuatan Fungsi Hapus Resep

Fungsi hapus resep merupakan fungsi yang berguna untuk melakukan penghapusan data resep pada *database website*.

Berikut adalah langkah pembuatan fungsi hapus resep :

1. Buat berkas baru bernama “hapus_resep.php” di dalam direktori “content”.



Gambar 6. 31 Membuat Berkas admin/content/hapus_resep.php

2. Masukkan kode berikut pada berkas `hapus_resep.php`.

```
admin/content/hapus_resep.php
<?php
    if(!defined("INDEX")) die("");
    if(!empty($_GET["id_resep"])){
        $get_data = $conndB->query("SELECT
* FROM RESEP WHERE ID_RESEP =
'".$_GET['id_resep']."' ");
        $data = $get_data->fetchArray();
        $conndB->exec("PRAGMA
foreign_keys=on");
        $delete = $conndB->query("DELETE
FROM RESEP WHERE ID_RESEP =
'".$_data['ID_RESEP']."' ");
        if($delete){
            $new_loc =
$_SERVER['DOCUMENT_ROOT']."/assets/images
/";
            unlink($new_loc.$data['GAMBAR']);
            $update_data = $conndB-
>query("UPDATE PERMINTAANRESEP SET
ID_STATUS = 'sts01' WHERE ID_PERMINTAAN =
'".$_data["ID_PERMINTAAN']."' ");
            if($update_data){
                $_SESSION["pesan"] = "<span
style='color:blue;'>Berhasil :</span>
Resep dengan nama
".$_data["NAMA RESEP"]."[".$_GET["id_resep
"]."] berhasil dihapus";
                echo "<meta http-
equiv='refresh'
content='0;url=?page=resep'>";
            }
        }
    }
```



```

        $conndB->exec("PRAGMA
foreign_keys=off");
    }
?>

```

Tabel 6. 27 Kode Untuk Berkas *admin/content/hapus_resep.php*

Kode pada berkas “hapus_resep.php” diatas berfungsi untuk:

- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa fungsi hapus resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas hapus_resep.php.
- o **Membuat fungsi hapus resep** : Berguna untuk menerima data *id* resep dari *parameter url*, kemudian menghapus resep dengan *id* tersebut pada *database*.
- o **Membuat fungsi hapus gambar** : Berfungsi untuk menghapus berkas gambar resep yang sebelumnya tersimpan pada direktori “assets/images”.
- o **Membuat fungsi *update* status permintaan resep** : Berfungsi untuk mengubah status dari permintaan resep yang sebelumnya “telah ditambahkan” menjadi “belum ditambahkan”. Fungsi ini hanya akan berjalan jika resep

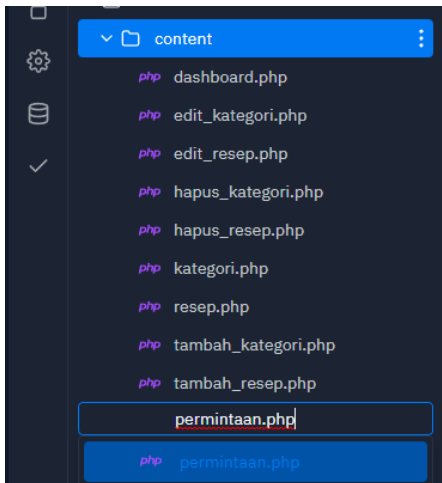
yang akan dihapus, merupakan resep yang berasal dari permintaan pengunjung.

6.3.13 Pembuatan *Interface* Daftar Permintaan Resep

Interface daftar permintaan resep merupakan *interface* yang berisi daftar permintaan resep apa saja yang ada pada *database website*.

Berikut adalah langkah pembuatan *interface* daftar permintaan resep:

1. Buat berkas baru bernama “permintaan.php” di dalam direktori “content”.



Gambar 6. 32 Membuat Berkas admin/content/permintaan.php

2. Masukkan kode berikut pada berkas permintaan.php.

```
admin/content/permintaan.php
<?php
    if(!defined("INDEX")) die("");
    $conndB->query("insert or ignore into
STATUSPERMINTAAN values('sts01','Belum
Ditambahkan'),('sts02','Telah
Ditambahkan'),('sts03','Duplikat)");

    if(!empty($_POST['filter'])){
        $_SESSION['filter'] =
$_POST['filter'];
    }

    if(empty($_SESSION['filter'])){
        $_SESSION['filter'] = "sts01";
    }

    $get_data = $conndB->query("SELECT *
FROM PERMINTAANRESEP as pr LEFT JOIN
KATEGORIRESEP as kat ON pr.ID_KATEGORI =
kat.ID_KATEGORI LEFT JOIN
STATUSPERMINTAAN as sr ON pr.ID_STATUS =
sr.ID_STATUS WHERE pr.ID_STATUS =
'".$_SESSION['filter']."' AND
NAMA_PERMINTAAN LIKE
'%" . $_POST['search'] . "%' ORDER BY
ID_STATUS ASC, TANGGAL_PERMINTAAN DESC");

?>
<div class="col-md-11">
    <div class="content-head">
        <h1>Data Permintaan Resep</h1>
    </div>
```

```

<div class="p-1 bg-info offset-md-6 d-
flex">
    <h5 class="m-1 p-1">Filter :</h5>
    <form method="POST" action="">
        <div class=""<?php
if($_SESSION['filter'] == 'sts01'){ echo
'bg-warning';}?>">
            <button class="btn btn-primary m-
1 p-1 <?php if($_SESSION['filter'] ==
'sts01'){ echo 'active';}?>"
type="submit" name="filter" value="sts01"
>Belum Ditambahkan</button>
        </div>
    </form>
    <form method="POST" action="" >
        <div class=""<?php
if($_SESSION['filter'] == 'sts02'){ echo
'bg-warning';}?>">
            <button class="btn btn-primary m-
1 p-1 <?php if($_SESSION['filter'] ==
'sts02'){ echo 'active';}?>"
type="submit" name="filter" value="sts02"
>Telah Ditambahkan</button>
        </div>

    </form>
</div>
<div class="content-button">
    <p>
        <?php
            echo $_SESSION["pesan"];
            unset($_SESSION["pesan"]);
        ?>
    </p>
</div>
<div class="content-table">
    <table class="table table-striped
table-hover">

```

```

<thead>
  <tr>
    <th>No</th>
    <th>Nama Resep</th>
    <th>Kategori</th>
    <th>Catatan</th>
    <th>Status</th>
    <th>Aksi</th>
  </tr>
</thead>
<tbody>
  <?php
    $i = 0;
    while($data = $get_data-
>fetchArray()) {
      $i++
      ?>
      <tr>
        <td><?php echo $i;?></td>
        <td><?php echo
$data["NAMA_PERMINTAAN"];?></td>
        <td>
          <?php
            if ($data["ID_KATEGORI"] ==
"" ) {
              echo "Tidak terkategori";
            } else {
              echo
$data["NAMA_KATEGORI"];
            }
          ?>
        </td>
        <td><?php echo
$data["CATATAN_PERMINTAAN"];?></td>
        <td><?php echo
$data["NAMA_STATUS"];?></td>
        <td><?php

```

```

                                if($data["ID_STATUS"] !=
"sts02"){
                                echo '<form
method="POST"
action="?page=tambah_resep" class="p-1">
                                <input
type="hidden" name="id_permintaan"
value="'. $data["ID_PERMINTAAN"].' ">
                                <button
class="btn btn-warning"
type="submit">Tambahkan ke Resep</button>
                                </form>
                                <form
method="POST"
action="?page=permintaan_duplikat"
class="p-1">
                                <input
type="hidden" name="id_permintaan"
value="'. $data["ID_PERMINTAAN"].' ">
                                <input
type="hidden" name="nama_permintaan"
value="'. $data["NAMA_PERMINTAAN"].' ">
                                <button
class="btn btn-danger"
type="submit">Tandai Duplikat</button>
                                </form>';
                                }
                                ?>
                                </td>
                                </tr>
                                <?php
                                };
                                ?>
                                </tbody>
                                </table>
                                </div>
                                </div>

```

Tabel 6. 28 Kode Untuk Berkas admin/content/permintaan.php

Kode pada berkas “permintaan.php” diatas berfungsi untuk:

- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa konten pada *interface* daftar permintaan resep ini diakses melalui *index interface admin*, bukan langsung melalui berkas *permintaan.php*.
- o **Membuat fungsi *insert status permintaan*** : Karena disini kita tidak menyediakan *interface* untuk menambahkan status permintaan, maka kita akan harus membuat suatu fungsi yang dapat menambahkan status permintaan resep. Untuk status yang ditambahkan ada 3, yaitu “Belum Ditambahkan”, “Telah Ditambahkan”, dan “Duplikat”.
- o **Mengambil dan menampilkan data dari *database*** : Data nama, kategori, catatan dan status permintaan resep akan diambil dari database, kemudian ditampilkan pada *interface* ini (kecuali permintaan resep dengan status “Duplikat”).

o Memfilter data berdasarkan status permintaan :

Berfungsi untuk hanya menampilkan data permintaan resep yang sesuai dengan status permintaan yang ingin ditampilkan.

o Tombol tambahkan ke resep, dan tandai duplikat :

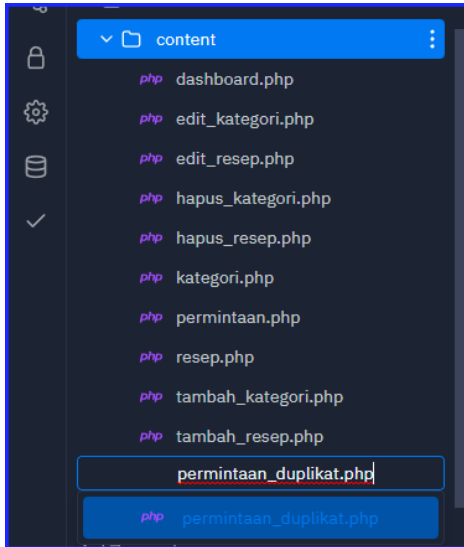
Berfungsi untuk dapat melakukan penambahan permintaan resep menjadi resep baru pada *websiste*(tombol ini akan tetap mengarah ke interface tambah resep, namun dengan mengirimkan data permintaan resep pengguna). sementara untuk tombol tandai duplikat, berfungsi untuk menandai suatu permintaan yang dianggap duplikat atau sama dengan lainnya.

6.3.14 Pembuatan Fungsi Tandai Permintaan Resep Sebagai Duplikat

Fungsi tandai permintaan duplikat merupakan fungsi yang berguna untuk dapat mengubah status permintaan resep menjadi duplikat, dika dirasa permintaan resep tersebut sama atau telah ditambahkan sebelumnya pada *database website*.

Berikut adalah langkah pembuatan fungsi tandai permintaan resep sebagai duplikat :

1. Buat berkas baru bernama “permintaan_duplikat.php” di dalam direktori “content”.



Gambar 6. 33 Membuat Berkas admin/content/permintaan_duplikat.php

2. Masukkan kode berikut pada berkas permintaan_duplikat.php.

admin/content/permintaan_duplikat.php
<pre><?php if (!defined("INDEX")) die(""); if (\$_SERVER["REQUEST_METHOD"] == "POST" AND !empty(\$_POST["id_permintaan"])) {</pre>

```

$update_data = $conndB-
>query("UPDATE PERMINTAANRESEP SET
ID_STATUS = 'sts03' WHERE ID_PERMINTAAN =
'".$_POST['id_permintaan']."'");
    if($update_data){
        $_SESSION["pesan"] = "<span
style='color:blue;'>Berhasil :</span>
Permintaan '".$_POST["nama_permintaan"]."
('".$_POST["id_permintaan"]."') berhasil
ditandai sebagai duplikat";
        echo "<meta http-equiv='refresh'
content='0;url=?page=permintaan'>";
    }
}
?>

```

Tabel 6. 29 Kode Untuk Berkas admin/content/permintaan_duplikat.php

Kode pada berkas “permintaan_duplikat.php” diatas berfungsi untuk:

- o **Membuat fungsi cek referensi *interface*** : Berfungsi untuk memastikan bahwa fungsi tandai permintaan resep sebagai duplikat ini diakses melalui *index interface admin*, bukan langsung melalui berkas permintaan_duplikat.php.

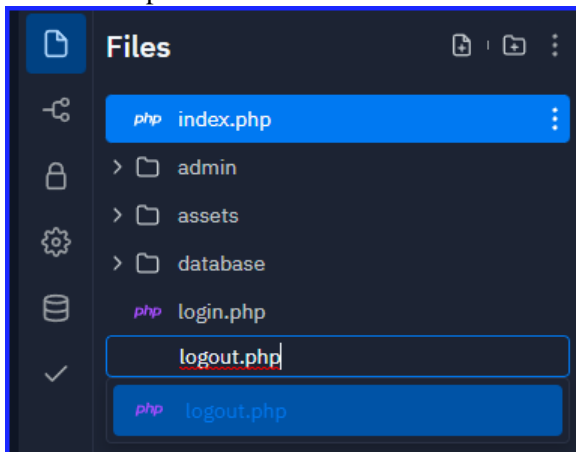
- o **Membuat fungsi *update* status permintaan resep** : Berfungsi untuk mengubah status dari permintaan resep yang sebelumnya “belum ditambahkan” menjadi “duplikat”.

6.3.15 Pembuatan Fungsi *Logout*

Fungsi *logout* merupakan fungsi yang mengakhiri menghapus sesi login pengguna pada *website*.

Berikut adalah langkah pembuatan fungsi *logout* :

1. Buat berkas baru bernama “logout.php” pada *home* direktori Replit.



Gambar 6. 34 Membuat Berkas *logout.php*

2. Masukkan kode berikut pada berkas *logout.php*.

logout.php
<pre><?php session_start(); session_destroy();</pre>

```
header("Location: login.php");  
?>
```

Tabel 6. 30 Kode Untuk Berkas *logout.php*

Kode pada berkas “logout.php” diatas berfungsi untuk:

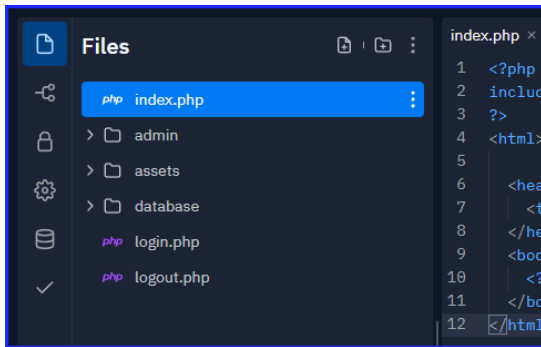
- o **Menghapus sesi pengguna** : Berfungsi untuk menghapus sesi pengguna yang telah ditambahkan pada saat pengguna berhasil melakukan *login*.

6.3.16 Pembuatan *Interface* Utama (Halaman Awal)

Interface utama (halaman awal) merupakan *interface* halaman utama website, pada halaman ini berisi tampilan kumpulan resep yang terdapat pada *database website*.

Berikut adalah langkah pembuatan *interface* utama (halaman awal) :

1. Buka berkas “index.php” pada *home* direktori Replit.



Gambar 6. 35 Membuka Berkas index.php

2. Hapus semua kode pada berkas “index.php”, kemudian ganti dengan kode dibawah ini.

```

index.php
<?php
    ob_start();
    include "database/conndB.php";
    if(!empty($_GET["res"])){
        $get_data_resep = $conndB->
        query("SELECT * FROM RESEP LEFT JOIN
        KATEGORI RESEP ON RESEP.ID_KATEGORI =
        KATEGORI RESEP.ID_KATEGORI WHERE ID_RESEP
        = '".$_GET["res"]."');
        $data_resep = $get_data_resep->
        fetchArray();
        if(empty($data_resep['0'])){
            $title = "Not Found";
        }else{
            $title =
            $data_resep["NAMA_RESEP"]." - Web Resep";
        }
        }else if (isset($_GET["search"])){

```

```

        $title = $_GET["search"]." - Web
Resep";
    }else if (!empty($_GET["cat"])){
        $get_data_resep = $conndB-
>query("SELECT * FROM KATEGORI RESEP AS
kat LEFT JOIN RESEP AS r ON
kat.ID_KATEGORI = r.ID_KATEGORI WHERE
kat.ID_KATEGORI = '".$_GET["cat"]."');
        $data_resep = $get_data_resep-
>fetchArray();
        if(empty($data_resep['0'])){
            $title = "Not Found";
        }else{
            $title =
            $data_resep["NAMA_KATEGORI"]." - Web
Resep";
        }
    }else if(isset($_POST['request'])){
        $title = "Request Resep - Web Resep";
    }else{
        if($_SERVER['REQUEST_URI'] != "/"){
            $title = "Not Found";
        }else{
            $title = "Web Resep";
        }
    }
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">

```

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfhHxaWutUpFmBp4vmVor"
crossorigin="anonymous">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"
integrity="sha512-KfkfwYDsLkIlwQp6LFnl8zNdLGxu9YAA1QvwINks4PhcElQSwqcyVLLD9aMhXdl3uQjoXtEKNosOWaZqXge10g==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
<link href="../../assets/style.css"
rel="stylesheet">
<title>
<?php
    echo $title;
?>
</title>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-primary navbar-blog w-100">
    <div class="container-fluid">
        <?php

if(isset($_SERVER['HTTP_REFERER'])) {
    if(!isset($_POST['backbutton'])) {
        echo '<form method="POST"
action="'.$_SERVER['HTTP_REFERER'].'"
class="d-flex">

```

```

        <button class="btn btn-info"
name="backbutton"
type="submit"></i>Kembali</button>
    </form>';
    }
}
?>
<ul class="navbar-nav mx-auto">
    <li class="nav-item p-1">
        <a class="nav-link text-white"
href="/">Home</a>
    </li>
    <div class="dropdown">
        <input type="checkbox" id="my-
dropdown" value="" name="my-checkbox">
        <label class="bg-warning"
for="my-dropdown"
            data-toggle="dropdown">
            Kategori <i class="fa-solid
fa-angle-down"></i>
        </label>
        <ul class="dropdown-menu">
            <?php
                $get_katdata = $conndB-
>query("SELECT * FROM KATEGORIRESEP");
                while($katdata =
$get_katdata->fetchArray()) {
                    $get_jumlahkat = $conndB-
>query("SELECT COUNT(*) as jumlah FROM
KATEGORIRESEP INNER JOIN RESEP ON
RESEP.ID_KATEGORI =
KATEGORIRESEP.ID_KATEGORI WHERE
KATEGORIRESEP.ID_KATEGORI =
'".$katdata['ID_KATEGORI']."'");
                    $jumlahkat =
$get_jumlahkat->fetcharray();
                ?>

```



```

        <li><a class="dropdown-
item" href="?cat=<?php echo
$katdata['ID_KATEGORI'];?>"><?php echo
$katdata['NAMA_KATEGORI'].('.$jumlahkat[
'jumlah'].')';?></a></li>
        <?php
        }
        ?>
    </ul>
</div>
</ul>
<form method="POST" action="/"
class="p-1">
    <button class="btn btn-warning"
name="request" type="submit">Request
Resep</button>
</form>
<form method="GET" action=""
class="d-flex">
    <input class="form-control me-2"
type="text" name="search" value="<?php
echo $_GET['search']?>"
placeholder="Search">
    <button class="btn btn-dark"
type="submit">Search</button>
</form>

</div>
</nav>
<div class="row min-vh-100">
    <?php
    if(!empty($_GET["res"])){
        if(empty($data_resep['0'])){
            include "notfound.php";
        }else{
            include "blog_konten.php";
        }
    }else if (isset($_GET["search"])){

```

```

        include "search.php";
    }else if (!empty($_GET["cat"])){
        if(empty($data_resep['0'])){
            include "notfound.php";
        }else{
            include "cat_list.php";
        }
    }else if(isset($_POST['request'])){
        include "request.php";

    }else{
        if($_SERVER['REQUEST_URI'] != "/"){
            include "notfound.php";
        }else{?>

        <div class="row featured bg-
light">

        <?php
            $get_resep_terbaru = $conndB-
>query("SELECT * FROM RESEP LEFT JOIN
KATEGORI RESEP ON RESEP.ID_KATEGORI =
KATEGORI RESEP.ID_KATEGORI ORDER BY
TANGGAL DESC LIMIT 4 ");
            while($resep_terbaru =
$get_resep_terbaru->fetchArray()){
                ?>
                <div class="col-lg-3 col-
md-6 col-sm-6 mx-auto p-3">
                    <div class="text-center">
                        " style="width: 250px; height: 250px;"
class="w-100" />
                        <a href="?res=<?php
echo $resep_terbaru['ID_RESEP'] ?>">
                            <div class="mask"
style="background-color: rgba(0, 0, 250,
0.7)">

```

```

                                <div class="d-flex
justify-content-center align-items-center
h-100">
                                <h4 class="text-
white mb-0"><?php echo
$resep_terbaru['NAMA_RESEP']?></h4>
                                <p class="badge
text-white"><?php echo
$resep_terbaru['NAMA_KATEGORI']?></p>
                                </div>
                                </div>
                                <div class="hover-
overlay">
                                <div class="mask"
style="background-color: rgba(251, 251,
251, 0.2)"></div>
                                </div>
                                </a>
                                </div>
                                </div>
                                <?php
                                }

                                ?>
                                </div>

                                <!-- #KATEGORI1 -->
                                <div class="row kategori bg-dark
bg-opacity-75 p-2 mx-auto m-4">
                                <?php
                                    $get_katname = $conndB-
>query("SELECT NAMA_KATEGORI FROM
KATEGORI RESEP AS kat WHERE
kat.ID_KATEGORI = 'kat01'");
                                    $katname = $get_katname-
>fetchArray();
                                ?>

```

```

        <div class="col-md-3 bg-primary
text-center p-4 h-25 bg-opacity-75">
            <h4 class="bg-warning" ><?php
echo $katname['NAMA_KATEGORI']?></h4>
        </div>
        <div class="col-md-9 bg-primary
p-3">
            <?php
                $get_dataresep = $conndB-
>query("SELECT * FROM KATEGORI RESEP AS
kat INNER JOIN RESEP AS r ON
kat.ID_KATEGORI = r.ID_KATEGORI WHERE
kat.ID_KATEGORI = 'kat01' LIMIT 3");
                while($data_resep =
$get_dataresep->fetchArray()) {
                    ?>
                    <a href="?res=<?php echo
$data_resep['ID_RESEP']?>" style="text-
decoration: none;">
                        <div class="row bg-
primary">

                            <div class="col-md-5
bg-light p-2" style="border:1px solid
#ddd;">
                                
" alt="gambar resep" class="img-fluid">
                                </div>
                                <div class="col-md-7
bg-light p-2 " style="border:1px solid
#ddd">
                                    <div class="bg-
primary text-center text-white">
                                        <h2><?php echo
$data_resep['NAMA_RESEP']?></h2>
                                    </div>

```

```

                                <div class="h-75
deskripsi-resep p-2">
                                <h5 class="text-
muted"><?php echo
$data_resep['DESKRIPSI']?></h5>
                                </div>

                                </div>
                                </div>
                                </a>
                                <?php
                                }
                                ?>

                                </div>
                                </div>

                                <!-- #KATEGORI2 -->
                                <div class="row kategori bg-info
bg-opacity-75 p-2 mx-auto m-4">
                                <?php
                                    $get_katname = $conndB-
>query("SELECT NAMA_KATEGORI FROM
KATEGORIRESEP AS kat WHERE
kat.ID_KATEGORI = 'kat02'");
                                    $katname = $get_katname-
>fetchArray();
                                    ?>
                                    <div class="col-md-3 bg-primary
text-center p-4 h-25 bg-opacity-75">
                                        <h4 class="bg-warning" ><?php
echo $katname['NAMA_KATEGORI']?></h4>
                                    </div>
                                    <div class="col-md-9 bg-dark p-
3">

                                        <?php
                                            $get_dataresep = $conndB-
>query("SELECT * FROM KATEGORIRESEP AS

```

```

kat INNER JOIN RESEP AS r ON
kat.ID_KATEGORI = r.ID_KATEGORI WHERE
kat.ID_KATEGORI = 'kat02' LIMIT 3");
        while($data_resep =
$get_dataresep->fetchArray()) {
            ?>
            <a href="?res=<?php echo
$data_resep['ID_RESEP']?>" style="text-
decoration: none;">
                <div class="row bg-
primary">
                    <div class="col-md-5
bg-light p-2" style="border:1px solid
#ddd;">
                        
" alt="gambar resep" class="img-fluid">
                    </div>
                    <div class="col-md-7
bg-light p-2 " style="border:1px solid
#ddd">
                        <div class="bg-
primary text-center text-white">
                            <h2><?php echo
$data_resep['NAMA_RESEP']?></h2>
                        </div>
                        <div class="h-75
deskripsi-resep p-2">
                            <h5 class="text-
muted"><?php echo
$data_resep['DESKRIPSI']?></h5>
                        </div>
                    </div>
                </div>
            </a>

```

```

        <?php
        }
    ?>

    </div>
</div>
<?php
}
}?>

<div class="row m-0">
    <footer class="footer mt-auto">
        <p>Copyright &copy;
190411100121</p>
    </footer>
</div>
</div>
</body>
</html>

```

Tabel 6. 31 Kode Untuk Berkas *index.php* (Kode Halaman Utama)

Kode pada berkas “index.php” diatas berfungsi untuk:

o **Mengambil dan menampilkan data dari *database* :**

Data nama, deskripsi, kategori, dan gambar resep akan diambil dari database serta direktori gambar “assets/images”, kemudian ditampilkan pada *interface* ini, baik berdasarkan resep terbaru (semua kategori), maupun spesifik berdasarkan kategori tertentu yang ingin ditampilkan.

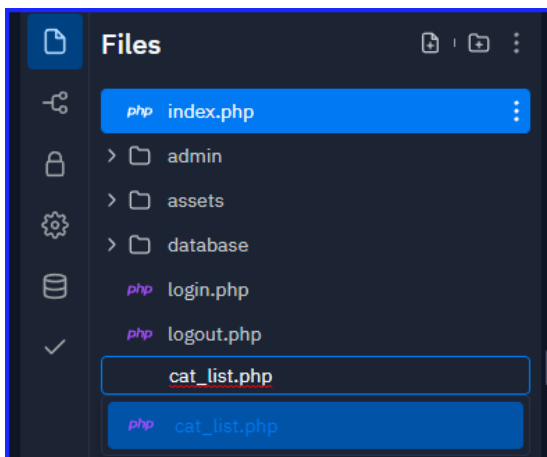
- o **Membuat Form Pencarian** : Berfungsi untuk memudahkan pengunjung dalam mencari nama resep yang diinginkan, sesuai dengan data resep yang ada pada *database* website.
- o **Membuat *navbar menu*** : Berfungsi untuk membuat navigasi yang dapat mengarah ke *interface* pengunjung lainnya dan kemudian ditampilkan di *interface* ini.
- o **Menampilkan isi konten *interface admin* lainnya** : Berguna untuk menampilkan konten dari *interface* pengunjung lainnya pada *interface* ini.

6.3.17 Pembuatan *Interface* Daftar Resep Berdasarkan Kategori

Interface daftar resep berdasarkan kategori merupakan *interface* yang berisi daftar atau kumpulan resep berdasarkan kategori tertentu.

Berikut adalah langkah pembuatan *interface* daftar resep berdasarkan kategori :

1. Buat berkas baru bernama “cat_list.php” pada *home* direktori Replit.



Gambar 6. 36 Membuat Berkas *cat_list.php*

2. Masukkan kode berikut pada berkas *cat_list.php*.

cat_list.php

```
<div class="row justify-content-center  
align-items-center p-3">  
  <div class="col-md-10 p-4">  
    <?php  
      if(empty($_GET["page"]) OR  
$_GET["page"] < 1){
```

Pembuatan Website Dinamis Berbasis PHP dan SQLite Database Menggunakan Replit

Cloud-Service

184

```

        $offset = 0;
    }else{
        $offset = $_GET["page"]*5;
    }
    $get_data_resep = $conndB->
query("SELECT * FROM KATEGORI RESEP AS
kat INNER JOIN RESEP AS r ON
kat.ID_KATEGORI = r.ID_KATEGORI WHERE
kat.ID_KATEGORI = '". $_GET["cat"]." '
LIMIT 5 OFFSET '". $offset. "'");
    while($data_resep =
$get_data_resep->fetchArray()){
        ?>
        <a href="?res=<?php echo
$data_resep['ID_RESEP']?>" style="text-
decoration: none;">
            <div class="row bg-info bg-
opacity-75 p-2">
                <div class="col-md-5 bg-light
p-2" style="border:1px solid
#ddd;">
                    
" alt="gambar resep" class="img-fluid">
                    </div>
                    <div class="col-md-7 bg-light
p-2 " style="border:1px solid
#ddd">
                        <div class="bg-primary
text-center text-white">
                            <h2><?php echo
$data_resep['NAMA_RESEP']?></h2>
                            </div>
                            <div class="h-75 deskripsi-
resep p-2">

```

```

        <h5 class="text-
muted"><?php echo
$data_resep['DESKRIPSI']?></h5>
    </div>
</div>
</div>
</a>
<?php
}
    $get_data_kategori = $conndB-
>query("SELECT * FROM KATEGORI RESEP AS
kat INNER JOIN RESEP AS r ON
kat.ID_KATEGORI = r.ID_KATEGORI WHERE
kat.ID_KATEGORI = '". $_GET["cat"]."' ");
    $data_kategori =
    $get_data_kategori->fetchArray();
    if(empty($data_kategori['0'])){
        $get_data_kategori = $conndB-
>query("SELECT * FROM KATEGORI RESEP AS
kat LEFT JOIN RESEP AS r ON
kat.ID_KATEGORI = r.ID_KATEGORI WHERE
kat.ID_KATEGORI = '". $_GET["cat"]."' ");
        $data_kategori =
        $get_data_kategori->fetchArray();
        ?>
        <div class="row bg-info bg-
opacity-75 p-2">
            <div class="row bg-primary
text-center text-white h-25">
                <h2>Data Resep Belum
Tersedia</h2>
            </div>
            <div class="row h-75 deskripsi-
resep p-2 text-center ">
                <h5 class="h-100 text-
muted"><p></p>Data resep dengan kategori
(<?php echo
$data_kategori['NAMA_KATEGORI']?>) yang

```

```

sedang Anda cari belum terdapat pada
database.</p><p> Anda dapat melakukan
request resep dengan kategori ini dengan
mengklik tombol di bawah ini</p></h5>
</div>
<div class="row col-md-4 col-
md-offset-4 mx-auto">
    <form method="POST"
action="/" class="d-flex">
        <button class="btn btn-
primary" name="request"
type="submit">Request Resep</button>
    </form>
</div>
</div>
<?php
}
?>
</div>
</div>
<div class="row">
    <ul class="pagination justify-content-
center align-items-center p-3">
        <?php
            $get_data = $conndB->query("SELECT
* FROM KATEGORI RESEP AS kat LEFT JOIN
RESEP AS r ON kat.ID_KATEGORI =
r.ID_KATEGORI WHERE kat.ID_KATEGORI =
'".$_GET["cat"]." ' LIMIT 5 OFFSET
'".$offset+5)."");
            $data = $get_data->fetchArray();
            if(empty($_GET['page'])) {
                if(!empty($data['0'])) {
                    echo '<li class="page-item"><a
class="page-link"
href="?cat='.$_GET["cat"].'&page=1">Next<
/a></li>';
                }
            }
        </?php
    </ul>
</div>

```

```

    }else{
        if($_GET['page'] < 2){
            echo '<li class="page-item"><a
class="page-link"
href="?cat='.$_GET["cat"].'">Prev</a></li
>';
        }else{
            echo '<li class="page-item"><a
class="page-link"
href="?cat='.$_GET["cat"].'&page='.$($_GET
["page"]-1).'">Prev</a></li>';
        }

        if(!empty($data['0'])){
            echo '<li class="page-item"><a
class="page-link"
href="?cat='.$_GET["cat"].'&page='.$($_GET
["page"]+1).'">Next</a></li>';
        }

    }
    ?>
</ul>
</div>

```

Tabel 6. 32 Kode Untuk Berkas *cat_list.php*

Kode pada berkas “cat_list.php” diatas berfungsi untuk:

o Mengambil dan menampilkan data dari *database* :

Data nama, deskripsi, dan gambar resep berdasarkan spesifik kategori tertentu akan diambil dari database serta direktori gambar “assets/images”, kemudian ditampilkan pada *interface* ini.

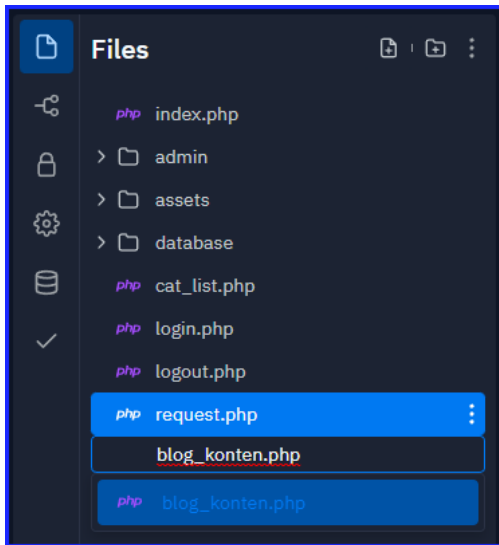
- o **Membuat fungsi paginasi** : Berfungsi untuk membatasi jumlah resep yang tampil pada suatu halaman, dan membagi keseluruhan data yang ada ke dalam beberapa halaman.

6.3.18 Pembuatan *Interface* Konten Resep

Interface konten resep merupakan *interface* yang berisi detail dari suatu resep yang berada pada *database website*. Detail ini meliputi nama, deskripsi, kategori, bahan, gambar, dan juga cara memasak makanan pada resep tersebut.

Berikut adalah langkah pembuatan *interface* konten resep :

1. Buat berkas baru bernama “blog_konten.php” pada *home* direktori Replit.



Gambar 6. 37 Membuat Berkas `blog_konten.php`

2. Masukkan kode berikut pada berkas `blog_konten.php`.

<code>blog_konten.php</code>
<pre> <div class="row justify-content-center align-items-center p-1"> <div class="col-md-7 bg-info bg- opacity-75"> <div class="row p-1"> <img class="gambar-konten-blog" src="<?php echo "/assets/images/" . \$data_resep['GAMBAR']?> " alt="gambar resep" class="img-fluid"> </div> <div class="row p-2"> </pre>

```

        <div class="bg-primary text-center
text-white">
            <h1><?php echo
$data_resep['NAMA_RESEP']?></h1>
            <span><?php echo
$data_resep['NAMA_KATEGORI']?></span>
        </div>
    </div>
    <div class="row">
        <div class="deskripsi-resep p-2">
            <h2 class="bg-
primary">Deskripsi</h2>
            <h5 class="text-muted"><?php echo
$data_resep['DESKRIPSI']?></h5>
        </div>
        <div class="row">
            <div class="h-75 bahan bg-warning
bg-opacity-75 p-2">
                <h2 class="bg-primary">Bahan</h2>
                <h5 class="text-muted">
                    <?php
                    $text =
nl2br($data_resep['BAHAN']);
                    function nl2pbahan($text) {
                        return "<ul><li>" .
str_replace("\n", "</li><li>", $text) .
"</li></ul>";
                    }
                    echo nl2pbahan($text);
                ?>
            </h5>
        </div>
    </div>
    <div class="row">
        <div class="h-75 bahan bg-info bg-
opacity-75 p-2">

```



```

        <h2 class="bg-primary">Cara
Memasak</h2>
        <h5 class="text-muted">
            <?php
                $text =
nl2br($data_resep['CARAMEMASAK']);
                function nl2pcarm($text) {
                    return "<ol><li>" .
str_replace("\n", "</li><li>", $text) .
"</li></ol>";
                }
                echo nl2pcarm($text);
            ?>
        </h5>
    </div>
</div>
</div>

```

Tabel 6. 33 Kode Untuk Berkas *blog_konten.php*

Kode pada berkas “blog_konten.php” diatas berfungsi untuk:

o **Mengambil dan menampilkan data dari *database* :**

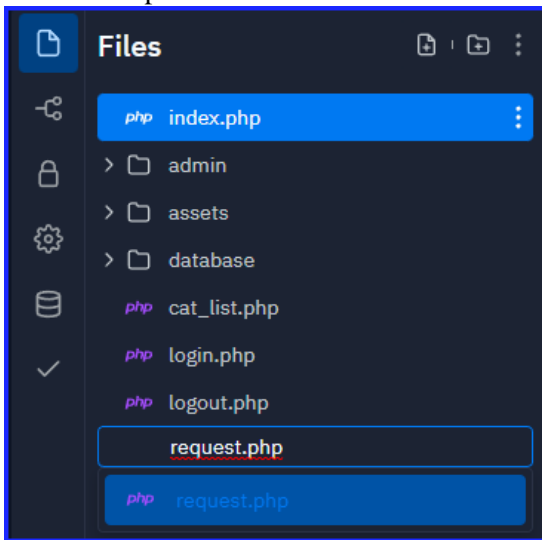
Data nama, deskripsi, kategori, gambar, bahan, dan cara memasak makanan pada resep akan diambil dari database serta direktori gambar “assets/images”, kemudian ditampilkan pada *interface* ini.

6.3.19 Pembuatan *Interface* Tambah Permintaan Resep

Interface tambah permintaan resep merupakan *interface* yang berisi form untuk melakukan penambahan data permintaan resep pada *database website*.

Berikut adalah langkah pembuatan *interface* tambah permintaan resep :

1. Buat berkas baru bernama “request.php” pada *home* direktori Replit.



Gambar 6. 38 Membuat Berkas *request.php*

2. Masukkan kode berikut pada berkas request.php.

```
request.php
<div class="row justify-content-center align-items-center p-1">
  <div class="col-md-9 p-4">
    <div class="row bg-info bg-opacity-75 p-2">
      <div class="row bg-primary text-center text-white">
        <h2>Form Request Resep</h2>
      </div>
      <div class="row deskripsi-resep col-md-9 mx-auto ">
        <form action="" method="POST" enctype="multipart/form-data">
          <div class="form-group">
            <label for="nama_resep_permintaan">Nama Resep</label>
            <input class="form-control" id="nama_resep_permintaan" name="nama_resep_permintaan" type="text" placeholder="Masukkan nama resep.." required>
          </div>
          <div class="form-group">
            <label for="kategori_resep_permintaan">Kategori Resep</label>
            <select class="form-control" id="kategori_resep_permintaan" name="kategori_resep_permintaan" required>
```

```

        <option value="">Silakan
Pilih Kategori Resep</option>
        <?php
            $get_data = $conndb-
>query("SELECT * FROM KATEGORI RESEP");
            while($data = $get_data-
>fetchArray()){
                echo "<option
value='".$data["ID_KATEGORI"]."'>".$data[
'NAMA_KATEGORI']. "</option>";
            }
        ?>
    </select>
</div>
    <div class="form-group">
        <label
for="catatan_resep_permintaan">Catatan</l
abel>
        <textarea class="form-
control" id="catatan_resep_permintaan"
name="catatan_resep_permintaan"
placeholder="Catatan resep untuk
penulis.."></textarea>
    </div>
    <div class="form-group">
        <input class="btn btn-
success" id="submit" name="request"
type="submit" value="Tambah">
        <a class="btn btn-danger"
href="<?php echo $_SERVER['HTTP_REFERER']
?>">Batal</a>
    </div>
</form>
<?php
    if($_SERVER["REQUEST_METHOD"]
== "POST" AND
!empty($_POST["nama_resep_permintaan"])){

```

```

        $get_data = $conndB-
>query("SELECT COUNT(*) as
jumlahpermintaan FROM PERMINTAANRESEP
WHERE NAMA_PERMINTAAN =
'".$_POST["nama_resep_permintaan"]." AND
CATATAN_PERMINTAAN =
'".$_POST["catatan_resep_permintaan"]."
AND ID_KATEGORI =
'".$_POST["kategori_resep_permintaan"]."
");
        $cek_data = $get_data-
>fetchArray();

if($cek_data['jumlahpermintaan'] == 0){
    $prefix = "pres";
    $get_data = $conndB-
>query("SELECT * FROM PERMINTAANRESEP
ORDER BY TANGGAL_PERMINTAAN DESC");
    $data = $get_data-
>fetchArray();
    if($data[0] == NULL){
        $prefix = "pres1";
    }else{
        $get_new_int =
intval(substr($data["ID_PERMINTAAN"],
4)+1);
        $prefix =
$prefix.$get_new_int";
    }
    $insert_data = $conndB-
>query("INSERT INTO PERMINTAANRESEP
VALUES('".$prefix."',
'".$_POST["kategori_resep_permintaan"]."
, 'sts01',
'".$_POST["nama_resep_permintaan"]." ,
'".$_POST["catatan_resep_permintaan"]." ,
strftime('%Y-%m-%d %H-%M-%S', 'now'))");
    if($insert_data){

```

```

        echo "<div class='alert
alert-success' role='alert'><span
style='color:blue;'>Berhasil :</span>
permintaan resep
('.$_POST['nama_resep_permintaan'].")
berhasil ditambahkan</div>";
    }
    }else {
        echo "<div class='alert
alert-danger' role='alert'><span
style='color:red;'>Gagal :</span>
permintaan resep
('.$_POST['nama_resep_permintaan'].")
sudah terdapat dalam database</div>";
    }
    }
    ?>
</div>
</div>
</div>
</div>

```

Tabel 6. 34 Kode Untuk Berkas request.php

Kode pada berkas “request.php” diatas berfungsi untuk:

- o **Membuat *post form*** : Berguna untuk mengirimkan data resep yang ingin ditambahkan oleh *admin*.
- o **Membuat fungsi unik *id* permintaan resep** : Berfungsi untuk membuat *id* permintaan resep yang unik dan berbeda dengan table lainnya pada *database* (menambahkan *prefix* pada *id* permintaan resep)

o **Membuat fungsi *insert* data permintaan resep :**

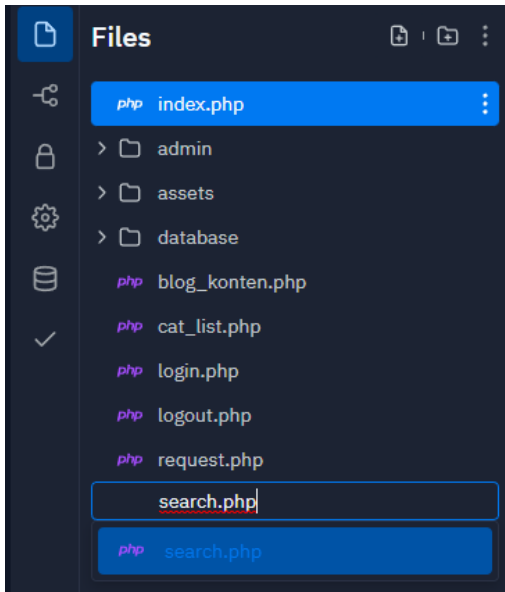
Berfungsi untuk menambahkan data permintaan resep yang telah diterima dari *post form* beserta *id* permintaan resep unik yang telah dibuat sebelumnya, ke dalam *database website*.

6.3.20 Pembuatan *Interface* Hasil Pencarian

Interface hasil pencarian merupakan *interface* yang berisi daftar atau kumpulan resep hasil *input* pengunjung pada form pencarian.

Berikut adalah langkah pembuatan *interface* hasil pencarian :

1. Buat berkas baru bernama “search.php” pada *home* direktori Replit.



Gambar 6. 39 Membuat Berkas *search.php*

2. Masukkan kode berikut pada berkas *search.php*.

search.php
<pre><div class="row justify-content-center align-items-center p-1"> <div class="col-md-10 p-4"> <?php if(!empty(\$_GET['search'])) { \$get_data_resep = \$conndB- >query("SELECT * FROM RESEP LEFT JOIN KATEGORIRESSEP ON RESEP.ID_KATEGORI = KATEGORIRESSEP.ID_KATEGORI WHERE</pre>


```

NAMA RESEP LIKE
'%" .$_GET["search"] ."% '");
    while($data_resep =
$get_data_resep->fetchArray()){
        ?>
        <a href="?res=<?php echo
$data_resep['ID_RESEP']?>" style="text-
decoration: none;">
            <div class="row bg-info bg-
opacity-75 p-2">
                <div class="col-md-5 bg-
light p-2" style="border:1px solid
#ddd;">
                    
" alt="gambar resep" class="img-fluid">
                    </div>
                    <div class="col-md-7 bg-
light p-2" style="border:1px solid
#ddd">
                        <div class="bg-primary
text-center text-white">
                            <h2><?php echo
$data_resep['NAMA_RESEP']?></h2>
                            <span class="bg-
warning" ><?php echo
$data_resep['NAMA_KATEGORI']?></span>
                        </div>
                        <div class="h-75
deskripsi-resep p-2">
                            <h5 class="text-
muted"><?php echo
$data_resep['DESKRIPSI']?></h5>
                        </div>
                    </div>
                </div>
            </a>

```

```

        <?php
        }
        $get_data_resep = $conndB-
>query("SELECT * FROM RESEP WHERE
NAMA_RESEP LIKE
'%.${_GET["search"]}.%'");
        $data_resep = $get_data_resep-
>fetchArray();
        }
        if(empty($data_resep['0']) OR
empty($_GET['search']) ){?>
        <div class="row bg-info bg-
opacity-75 p-2">
            <div class="row bg-primary
text-center text-white h-25">
                <h2>Data Tidak Ditemukan</h2>
            </div>
            <div class="row h-75 deskripsi-
resep p-2 text-center ">
                <h5 class="h-100 text-
muted"><p></p>Data dengan kata kunci
(<?php echo $_GET['search']?) yang
sedang Anda cari tidak ditemukan.</p><p>
Silakan coba kembali dengan kata kunci
lain</p></h5>
            </div>
            <div class="row col-md-4 col-
md-offset-4 mx-auto">
                <form method="GET"
action="" class="d-flex">
                    <input class="form-
control" type="text" name="search"
value="<?php echo $_GET['search']?"
placeholder="Search">
                    <button class="btn btn-
dark" type="submit">Search</button>
                </form>
            </div>

```

```

        </div>
    <?php
    }
    ?>
</div>
</div>

```

Tabel 6. 35 Kode Untuk Berkas *search.php*

Kode pada berkas “search.php” diatas berfungsi untuk:

o **Mengambil dan menampilkan data dari *database* :**

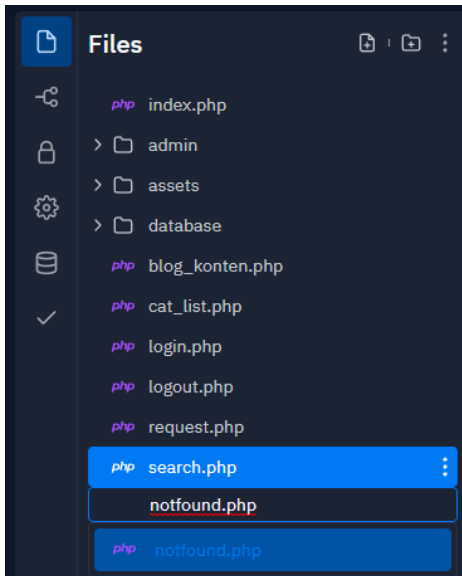
Data nama, deskripsi, kategori dan gambar resep berdasarkan berdasarkan data yang *diinputkan* pengguna pada form penarian, akan diambil dari database serta direktori gambar “assets/images”, kemudian ditampilkan pada *interface* ini.

6.3.21 Pembuatan *Interface* Halaman Tidak Ditemukan

Interface halaman tidak ditemukan merupakan *interface* yang berisi tampilan atau tulisan halaman tidak ditemukan, interface ini berfungsi untuk menginformasikan pengunjung bahwa halaman yang ingin dibuka tidak terdapat pada direktori ataupun *database website*.

Berikut adalah langkah pembuatan *interface* halaman tidak ditemukan :

1. Buat berkas baru bernama “notfound.php” pada *home* direktori Replit.



Gambar 6. 40 Membuat Berkas *notfound.php*

2. Masukkan kode berikut pada berkas *notfound.php*.

notfound.php

```
<div class="d-flex align-items-center  
justify-content-center vh-100 bg-dark">  
  <h1 class="display-4 fw-bold text-white  
text-center">404 <br> Halaman Tidak  
Ditemukan</h1><br>  
</div>
```

Tabel 6. 36 Kode Untuk Berkas *notfound.php*

Kode pada berkas “search.php” diatas berfungsi untuk:

o **Menampilkan tampilan halaman tidak ditemukan :**

Berfungsi untuk memberikan informasi kepada pengunjung bahwa halaman atau interface yang ingin dikunjungi tidak ditemukan, dengan menampilkan tulisan “404 halaman tidak ditemukan”.

6.3.22 Lampiran Kode CSS Tambahan & Berkas *Project*

Untuk mendapatkan tampilan yang lebih menarik, disini penulis menyediakan kode CSS tambahan (External CSS) yang dapat kalian gunakan pada *interface* yang telah kita buat sebelumnya, kalian juga dapat memodifikasi kode ini untuk mendapatkan tampilan yang sesuai dengan selera kalian. Untuk kodenya, kalian bisa melihatnya pada Lampiran 1. Modul CSS Pendukung (CSS *External*) .

Selain itu, disini penulis juga melampirkan berkas *project* web resep makanan yang telah dibuat pada *platform* Replit. Lampiran berkas ini bertujuan untuk mempermudah pembaca apabila ada yang merasa kesulitan dalam menyalin kode yang cukup banyak pada Perancangan Interface. Untuk melihat berkasnya, kalian dapat mengaksesnya melalui tautan yang ada pada Lampiran 2. Berkas *Project* Web Resep Makanan.

6.4 Pengujian Aplikasi

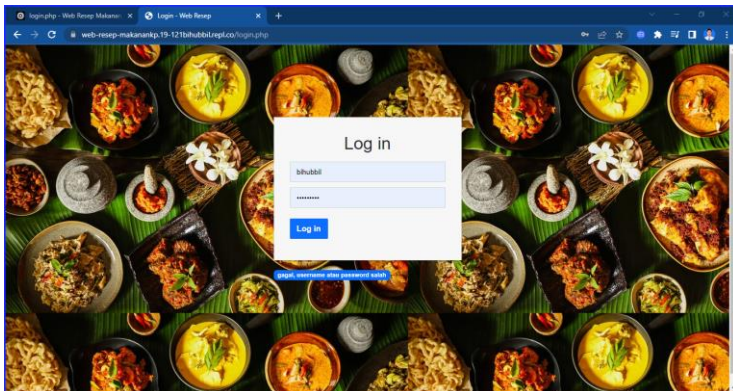
Setelah selesai merancang semua *interface* dan fungsi yang diperlukan untuk membuat web resep makanan, selanjutnya disini kita akan melakukan pengujian fungsi *interface* untuk memastikan semuanya dapat berjalan dan berfungsi dengan baik saat digunakan.

6.4.1 Pengujian Fungsi *Login Admin*

Pada pengujian fungsi *login admin* ini, pengujian akan dilakukan dengan menginputkan kredensial yang salah dan

benar. Pengujian ini berfungsi untuk mengetahui pengecekan kredensial untuk login sudah benar atau tidak.

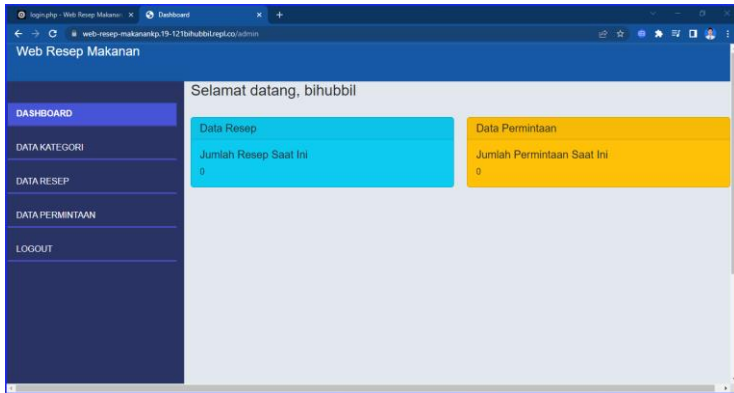
- *Menginputkan kredensial salah :*



Gambar 6. 41 Menginputkan Kredensial Salah

Saat mencoba melakukan login menggunakan kredensial yang salah, interface tersebut memberikan respon “gagal, username atau password salah”.

- Menginputkan kredensial benar :



Gambar 6. 42 Menginputkan Kredensial Benar

Namun saat mencoba melakukan login menggunakan kredensial yang benar, maka pengguna akan diarahkan ke halaman *dashboard* utama *admin*.

Sebagai informasi, berikut adalah kredensial *default admin* yang dapat digunakan untuk melakukan login :

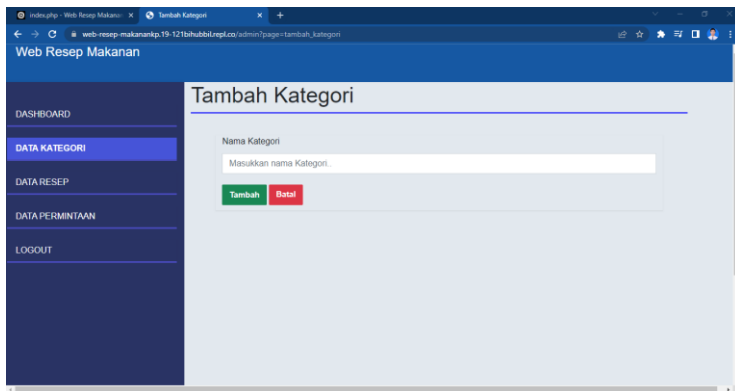
- Username : bihubbil
- Password : nim19-121

Dari kedua hasil diatas, dapat disimpulkan bahwa pengujian yang dilakukan sebelumnya berhasil dilakukan. serta fungsi login yang diuji dapat bekerja sesuai fungsinya.

6.4.2 Pengujian Fungsi Kelola Kategori Resep

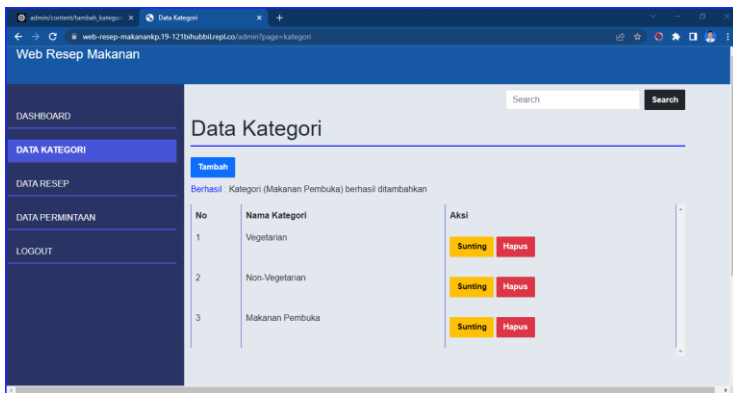
Pada pengujian fungsi kelola kategori resep ini, pengujian akan dilakukan dengan cara melakukan pengelolaan atau manajemen kategori resep, seperti menambahkan kategori resep baru, menyunting, serta menghapus kategori resep yang sudah ada.

- Menambahkan kategori resep



Gambar 6. 43 Menambahkan Kategori Resep

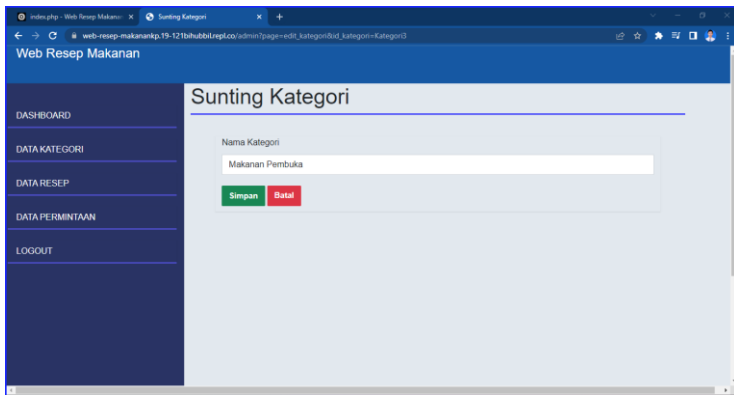
Pada pengujian fungsi tambah kategori resep, pengujian dilakukan dengan cara menambahkan 3 kategori resep baru. Untuk dapat menambahkan kategori resep baru, admin dapat melakukannya dengan cara mengklik tombol “tambah” yang berada pada menu “data kategori”.



Gambar 6. 44 Hasil Penambahan Kategori Resep

Dapat terlihat pada tampilan *interface* data kategori di atas, terdapat tambahan 3 kategori resep baru, hal ini berarti penambahan data kategori resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

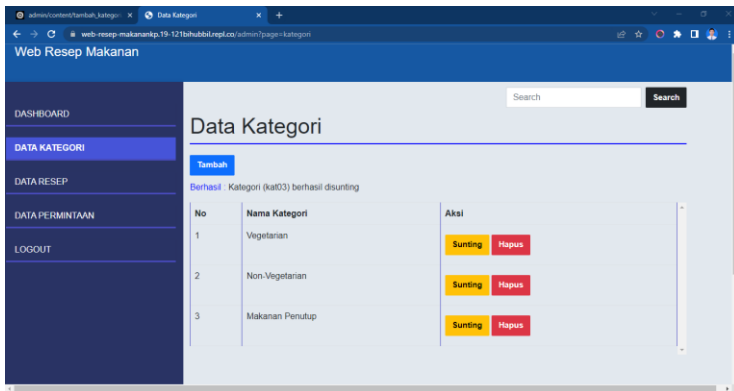
- Menyunting kategori resep



Gambar 6. 45 Menyunting Kategori Resep

Pada pengujian fungsi sunting kategori resep, pengujian dilakukan dengan cara menyunting salah satu kategori resep yang telah ditambahkan sebelumnya, disini kita akan mencoba untuk mengubah nama kategori resep “makanan pembuka” menjadi “makanan penutup”. Untuk dapat menyunting kategori resep, admin dapat melakukannya

dengan cara mengklik tombol “sunting” yang berada pada menu “data kategori”, sesuai dengan data kategori resep yang ingin disunting.

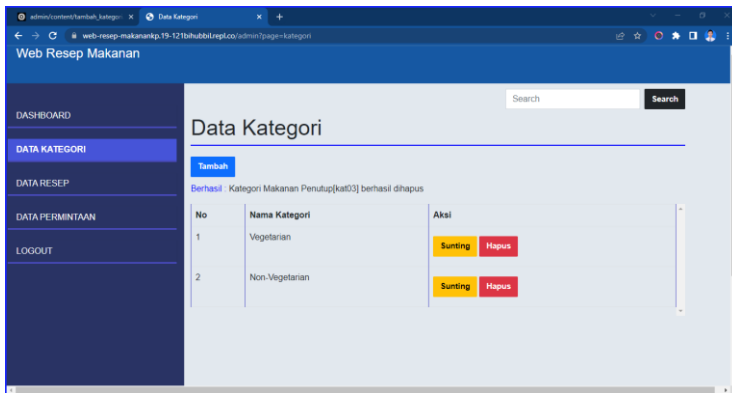


Gambar 6. 46 Hasil Sunting Kategori Resep

Dapat terlihat pada tampilan *interface* data kategori di atas, kategori resep yang sebelumnya memiliki nama “makanan pembuka”, berubah menjadi “makanan penutup”. Hal ini berarti penyuntingan data kategori resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

- Menghapus kategori resep

Pada pengujian fungsi hapus kategori resep, pengujian dilakukan dengan cara menghapus salah satu resep yang telah ditambahkan sebelumnya, disini kita akan mencoba menghapus kategori resep “makanan penutup”. Untuk dapat menghapus kategori resep, admin dapat melakukannya dengan cara mengklik tombol “hapus” yang berada pada menu “data kategori”, sesuai dengan data kategori resep yang ingin dihapus.



Gambar 6. 47 Hasil Hapus Kategori Resep

Dapat terlihat pada tampilan *interface* data kategori di atas, yang sebelumnya terdapat 3 kategori resep, kini berkurang menjadi 2 kategori resep . Hal ini berarti penghapusan data kategori resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

6.4.3 Pengujian Fungsi Kelola Resep

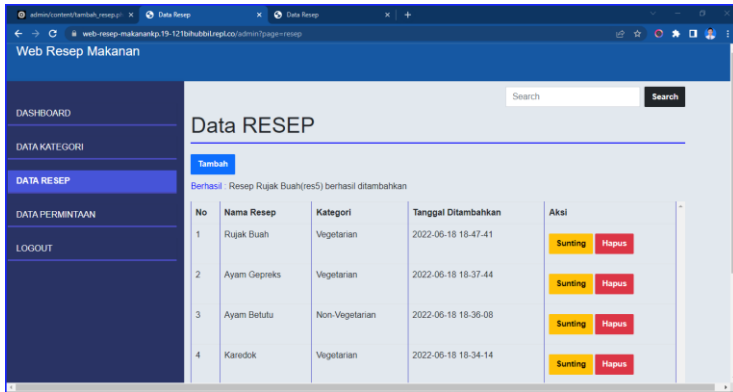
Pada pengujian fungsi kelola resep ini, pengujian akan dilakukan dengan cara melakukan pengelolaan atau manajemen resep, seperti menambahkan resep baru, menyunting, serta menghapus resep yang sudah ada.

- Menambahkan resep



Gambar 6. 48 Menambahkan Resep

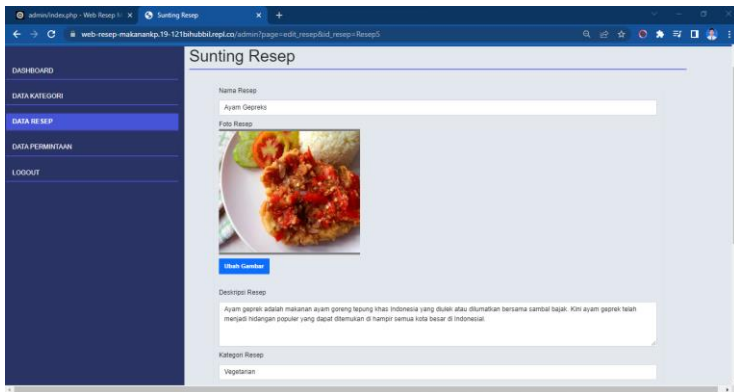
Pada pengujian fungsi tambah resep, pengujian dilakukan dengan cara menambahkan 5 resep baru. Untuk dapat menambahkan resep baru, admin dapat melakukannya dengan cara mengklik tombol “tambah” yang berada pada menu “data resep”.



Gambar 6. 49 Hasil Tambah Resep

Dapat terlihat pada tampilan *interface* data resep di atas, terdapat tambahan 5 resep baru, hal ini berarti penambahan data resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

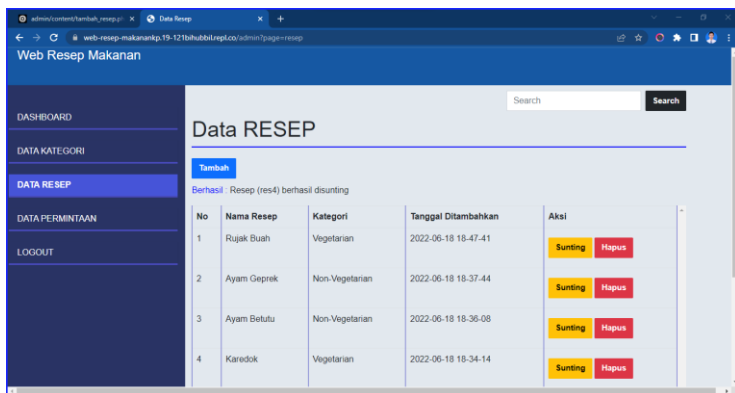
- Menyunting resep



Gambar 6. 50 Menyunting Resep

Pada pengujian fungsi sunting resep, pengujian dilakukan dengan cara menyunting salah satu resep yang telah ditambahkan sebelumnya, disini kita akan mencoba untuk menyunting “ayam gepreks” dengan mengubah namanya menjadi “ayam geprek” dan kategorinya menjadi “non-vegetarian”. Untuk dapat menyunting resep, admin dapat

melakukannya dengan cara mengklik tombol “sunting” yang berada pada menu “data resep”, sesuai dengan data resep yang ingin disunting.

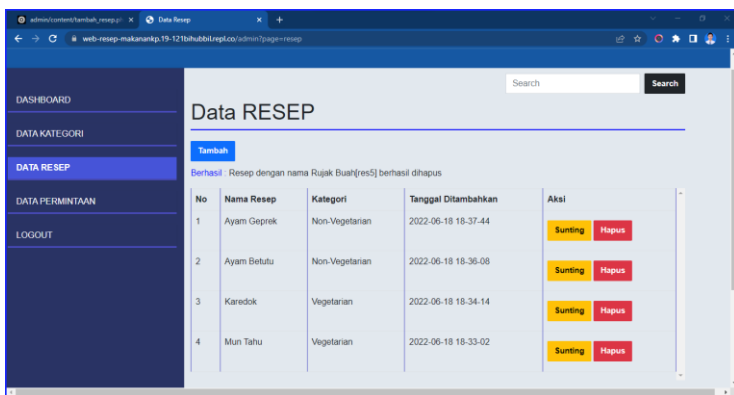


Gambar 6. 51 Hasil Sunting Resep

Dapat terlihat pada tampilan *interface* data resep di atas, resep yang sebelumnya memiliki nama “ayam gepreks”, berubah menjadi “ayam geprek”, kemudian yg sebelumnya berkategori “vegetarian”, berubah menjadi “non-vegetarian”. Hal ini berarti penyuntingan data resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

- Menghapus resep

Pada pengujian fungsi hapus resep, pengujian dilakukan dengan cara menghapus salah satu resep yang telah ditambahkan sebelumnya, disini kita akan mencoba menghapus resep “rujak buah”. Untuk dapat menghapus resep, admin dapat melakukannya dengan cara mengklik tombol “hapus” yang berada pada menu “data resep”, sesuai dengan data resep yang ingin dihapus.

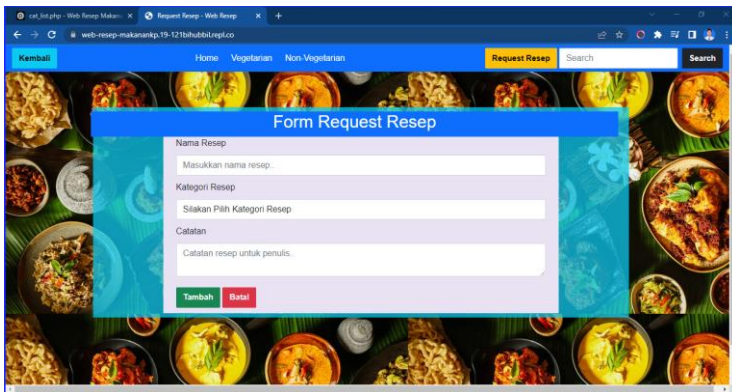


Gambar 6. 52 Hasil Hapus Resep

Dapat terlihat pada tampilan *interface* data resep di atas, kini resep “rujak buah” sudah tidak terdapat lagi pada

daftar tersebut . Hal ini berarti penghapusan data resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

6.4.3 Pengujian Fungsi Kirim Permintaan Resep Pengunjung



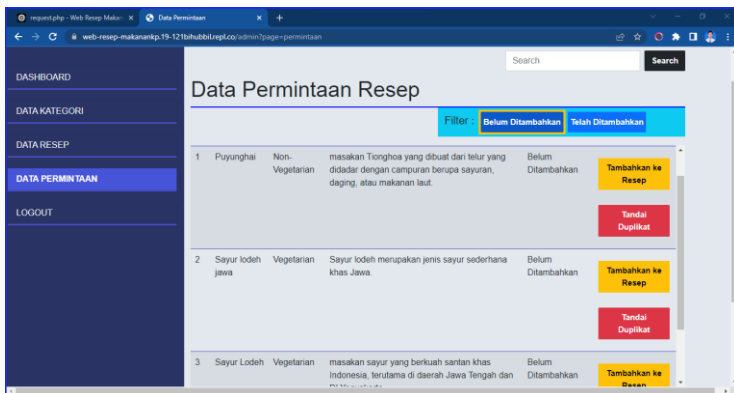
The screenshot displays a web browser window with the URL `web-resep-makananip.19-121bhubilrepl.it`. The page features a blue navigation bar with links: [Kembali](#), [Home](#), [Vegetarian](#), and [Non-Vegetarian](#). A yellow [Request Resep](#) button and a search bar are also present. The main content area is a modal titled 'Form Request Resep' with a light blue background. The form includes the following fields:

- Nama Resep**: A text input field with the placeholder 'Masukkan nama resep..'
- Kategori Resep**: A dropdown menu with the placeholder 'Silakan Pilih Kategori Resep'.
- Catatan**: A text area with the placeholder 'Catatan resep untuk penulis.'

At the bottom of the form are two buttons: a green [Tambah](#) button and a red [Batal](#) button. The background of the page is a collage of various food dishes.

Gambar 6. 53 Menambahkan Permintaan Resep

Pada pengujian fungsi kirim permintaan resep pengunjung, pengujian dilakukan dengan cara menambahkan 3 permintaan resep baru. Untuk dapat mengirimkan permintaan resep baru, pengunjung dapat melakukannya dengan cara mengklik tombol “request resep” yang berada pada *navbar* website(halaman pengunjung).



Gambar 6. 54 Hasil Penambahan Permintaan Resep

Dapat terlihat pada tampilan *interface* data permintaan resep di atas, terdapat tambahan 3 permintaan resep baru, hal ini berarti penambahan data permintaan resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

6.4.4 Pengujian Fungsi Kelola Daftar Permintaan Resep Pengunjung

Pada pengujian fungsi kelola permintaan resep pengunjung ini, pengujian akan dilakukan dengan cara melakukan pengelolaan atau manajemen permintaan resep, seperti menambahkan permintaan resep pengunjung menjadi resep baru, serta menandai permintaan resep yang dianggap sama sebagai duplikat.

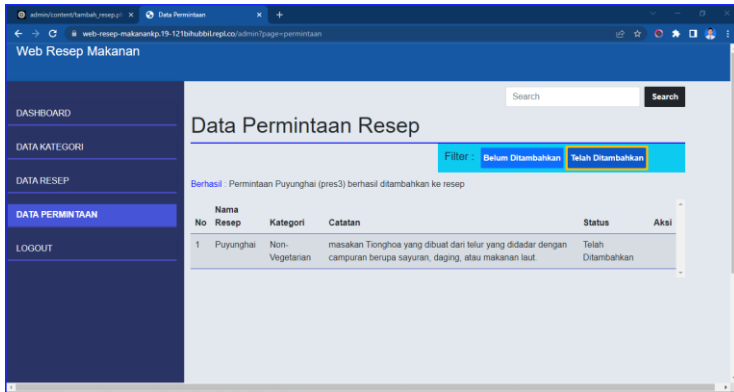
- Menambahkan permintaan resep pengunjung ke dalam resep baru



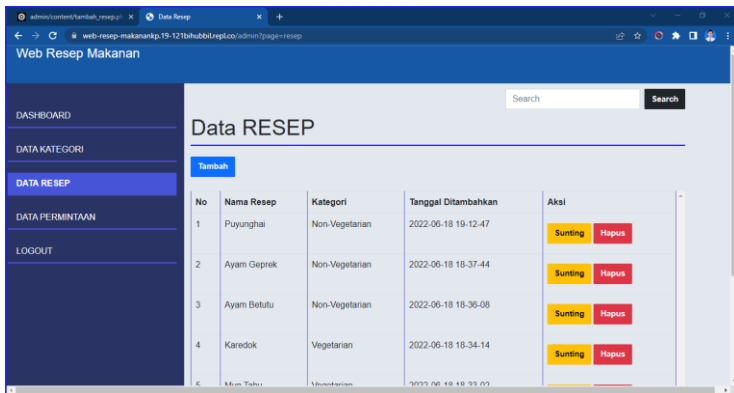
Gambar 6. 55 Menambahkan Permintaan Resep Menjadi Resep Baru

Pada pengujian fungsi tambah permintaan resep pengunjung ke dalam resep baru, pengujian dilakukan dengan cara menambahkan salah satu permintaan resep “puyunghai” kedalam resep baru.

Untuk dapat menambahkan resep baru, admin dapat melakukannya dengan cara mengklik tombol “tambahkan ke resep” yang berada pada menu “data permintaan”, sesuai dengan data permintaan yang ingin ditambahkan ke dalam resep baru.



Gambar 6. 56 Hasil Penambahan Permintaan Resep Menjadi Resep Baru (Data Permintaan Resep)



Gambar 6. 57 Hasil Penambahan Permintaan Resep Menjadi Resep Baru (Data Resep)

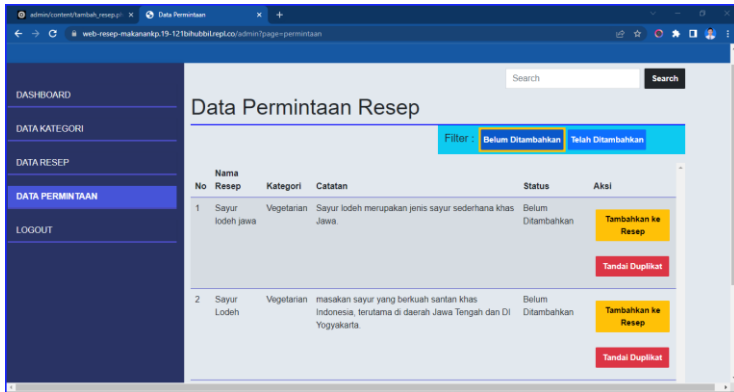
Dapat terlihat pada tampilan *interface* “data permintaan resep” dan “data resep” di atas, bahwa :

- o Pada interface data permintaan resep, permintaan resep “puyunghai”, yang status sebelumnya adalah “belum ditambahkan” berubah menjadi “telah ditambahkan”

- o Kemudian pada *interface* data resep, terdapat tambahan satu resep baru dengan nama resep “puyunghai”. Resep baru ini merupakan hasil penambahan dari permintaan resep pengunjung sebelumnya.

Kedua hal tersebut berarti mengindikasikan bahwa penambahan data permintaan resep ke dalam resep baru yang dilakukan sebelumnya, berhasil berjalan sesuai fungsinya.

- Menandai permintaan resep pengunjung sebagai duplikat



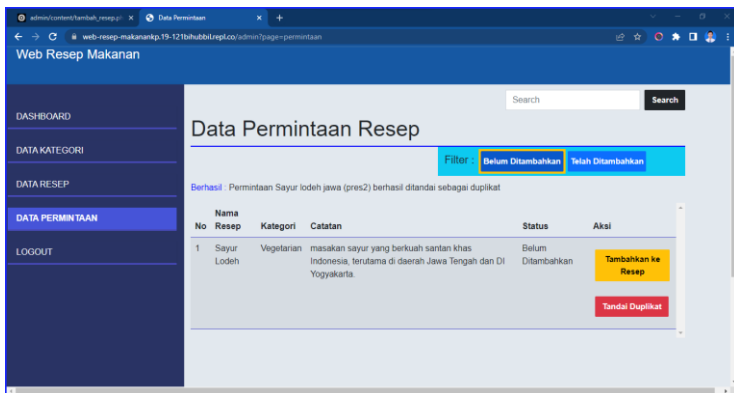
Gambar 6. 58 Menandai Permintaan Resep Sebagai Duplikat

Pada pengujian fungsi tandai permintaan resep pengunjung sebagai duplikat, pengujian dilakukan dengan cara menandai salah satu permintaan resep yang dianggap sama sebagai duplikat.

Sebagai permissalan, pada tampilan *interface* data permintaan resep di atas , antara “sayur lodeh” dan “sayur lodeh jawa” sebenarnya sama-sama berisi permintaan resep yang sama, namun dengan detil yang sedikit berbeda, jadi untuk menghindari penambahan resep yang sama sebanyak 2 kali atau lebih, *admin* dapat menandai

permintaan-permintaan yang dianggapnya duplikat dari permintaan sebelumnya yang telah ditambahkan. Oleh karena itu, disini kita dapat menandai “sayur lodeh jawa” sebagai duplikat permintaan.

Untuk dapat mendandai permintaan resep sebagai duplikat, admin dapat melakukannya dengan cara mengklik tombol “tandai sebagai duplikat” yang berada pada menu “data permintaan”, sesuai dengan data permintaan yang ingin ditandai sebagai duplikat.



Gambar 6. 59 Hasil Tandai Permintaan Resep Sebagai Duplikat

Dapat terlihat pada tampilan *interface* data permintaan resep di atas, permintaan resep “sayur lodeh jawa” yang sebelumnya ditandai sebagai duplikat, kini tidak lagi ditampilkan pada daftar permintaan resep. Hal ini berarti penambahan data permintaan resep yang dilakukan sebelumnya berhasil berjalan sesuai fungsinya.

DAFTAR PUSTAKA

- [1] Dicoding, "Apa itu Database? Contoh Produk dan Fungsinya," 16 September 2020. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-database/>. [Accessed 14 April 2022].
- [2] Replit, "Introduction to Replit," [Online]. Available: <https://docs.replit.com/>. [Accessed 20 June 2022].
- [3] Katalisnet, "Pengertian Website, Sejarah, dan Perkembangannya," 6 November 2020. [Online]. Available: <https://katalisnet.com/pengertian-dan-sejarah-website/>. [Accessed 6 April 2022].
- [4] Sqlite, "What Is SQLite?," [Online]. Available: <https://www.sqlite.org/index.html>. [Accessed 20 June 2022].
- [5] Andreasdan, "Pengertian HTML, PHP, CSS, Java Script dan XML," 6 December 2015. [Online]. Available: <https://andreasdan.com/html-php-css-javascript-xml/>. [Accessed 9 April 2022].
- [6] Minlab2, "7+ Jenis Website Berdasarkan Fungsinya, Mana Pilihanmu?," [Online]. Available: <https://bitlabs.id/blog/jenis-website/>. [Accessed 6 April 2022].

- [7] I. Mubarak, "Jenis-Jenis Website Berdasarkan Fungsi, Platform, dan Sifatnya," 22 August 2018. [Online]. Available: <https://www.niagahoster.co.id/blog/jenis-website/>. [Accessed 6 April 2022].

LAMPIRAN

Lampiran 1. Modul CSS Pendukung (CSS *External*)

assets/style.css

```
/* HALAMAN LOGIN */
.login-form {
    width: 350px;
    height : 400px;
    margin: 10% auto;
    font-size: 15px;
}
.login-form form {
    margin-bottom: 15px;
    background: #f7f7f7;
    box-shadow: 0px 2px 2px rgba(0, 0, 0,
0.3);
    padding: 30px;
}
.login-form h2 {
    margin: 0 0 15px;
}
.form-control, .btn {
    min-height: 38px;
    border-radius: 2px;
    margin: 10px auto;
}
.btn {
    font-size: 15px;
    font-weight: bold;
}
/* DASHBOARD */
body {
    margin: 0;
    padding: 0;
    font-family: sans-serif;
```

```

    background-image:
url(images/background.jpg);
    background-color: aquamarine;
}
.navbar-dashboard {
    background-color: #1658a3;
    height: 70px;
    width: 100%;
    padding-left: 30px;
    box-shadow: rgba(0, 0, 0, 0.24) 0px 3px
8px;
    font-weight: bolder;
    color: white;
}
.sidebar {
    background-color: #293364;
    display: block;
    min-height: 800px;
    max-height: 1500px;
    box-shadow: rgba(0, 0, 0, 0.24) 0px 3px
8px;
}
.sidebar ul {
    list-style-type: none;
    margin: 0;
    padding: 40px 2px;
    width: 100%;
    color: white;
}
.sidebar ul li {
    border-bottom: 3px solid rgb(77, 79, 199);
    margin-bottom: 20px;
    color: white;
}
.sidebar ul li a {
    display: block;
    color: white;
    padding: 8px 16px;

```



```

    text-decoration: none;
    box-shadow: rgba(0, 0, 0, 0.1) 0px 1px 2px
0px;
}
.sidebar ul li a.active {
    background-color: #4654d8;
    color: white;
    font-weight: bold;
}

.sidebar ul li a:hover:not(.active) {
    background-color: rgb(88, 108, 150);
    color: white;
}

.contents {
    background-color: rgb(226, 232, 238);
    margin-right: 0px;
}

.content-head {
    border-bottom: 2px solid blue;
}

.content-button {
    margin: 10px 0px;
}

.content-form {
    box-shadow: rgba(0, 0, 0, 0.1) 0px 1px 2px
0px;
    max-width: 90%;
}

.content-table {
    max-height: 450px;
    overflow-y: scroll;
}

.content-table td, .content-table th {

```

```

    border-color: rgb(77, 79, 199);
}
img {
    object-fit: fill;
}
.footer {
    background-color: #26538a;
    height: 50px;
    width: 100%;
    padding-top: 15px;
    box-shadow: rgba(0, 0, 0, 0.24) 0px 3px
8px;
    text-align: center;
    color: white;
    position: relative;
}
.editBtn {
    margin-bottom: 5%;
}
/* HALAMAN UTAMA */
.navbar-blog {
    height: 50px;
}
.featured {
    height: 300px;
    padding: 20px
    background-color: #40d8e2
}
.deskripsi-resep {
    background-color: #e9e2f3;
}
.gambar-list-kategori{
    width: 100%;
    height: 300px;
    object-fit: fill;
}
.gambar-konten-blog{
    width: 100%;

```

```

    height: 500px;
    object-fit: fill;
}

.dropdown {
    position: relative;
    display: inline-block;
    margin-top: 5px;
}

.dropdown > input[type="checkbox"] {
    position: absolute;
    left: -100vw;
}

.dropdown > label,
.dropdown > a[role="button"] {
    display: inline-block;
    padding: 6px 15px;
    color: #333;
    line-height: 1.5em;
    text-decoration: none;
    border: 1px solid #8c8c8c;
    cursor: pointer;
    -webkit-border-radius: 3px;
    -moz-border-radius: 3px;
    border-radius: 3px;
}

.dropdown > label:hover,
.dropdown > a[role="button"]:hover,
.dropdown > a[role="button"]:focus {
    border-color: #b5e40c;
}

.dropdown > ul {
    position: absolute;
    z-index: 999;

```

```

display: block;
left: -100vw;
top: calc(1.5em + 14px);
border: 1px solid #8c8c8c;
background: #fff;
padding: 6px 0;
margin: 0;
list-style: none;
width: 100%;
-webkit-border-radius: 3px;
-moz-border-radius: 3px;
border-radius: 3px;
-webkit-box-shadow: 0 3px 8px
rgba(0,0,0,.15);
-moz-box-shadow: 0 3px 8px
rgba(0,0,0,.15);
box-shadow: 0 3px 8px rgba(0,0,0,.15);
}

.dropdown > ul a {
display: block;
padding: 6px 15px;
text-decoration: none;
color: #333;
}

.dropdown > input[type="checkbox"]:checked ~
ul,
.dropdown > ul:target {
left: 0;
}

.dropdown a.close {
display: none;
}

```

```

.dropdown > ul:target ~ a.close {
  display: block;
  position: absolute;
  left: 0;
  top: 0;
  height: 100%;
  width: 100%;
  text-indent: -100vw;
  z-index: 1000;
}

```

Keterangan : Merupakan modul css pendukung yang berguna untuk mendapatkan tampilan visual website resep makanan, yang sama seperti pada saat proses pengujian.

Lampiran 2. Berkas *Project Web Resep Makanan*

Deskripsi	Isi
Lampiran berkas <i>project</i> web resep makanan (tautan ke berkas repl).	https://replit.com/@19-121Bihubbil/Web-Resep-MakananKP