



JS
=GO

TUTORIAL INTEGRASI API DALAM KATALOG MUSIK ONLINE



Aliffathur Muhammad Revan
Gilang Andhika Buwana
Syafrial Fachri Pane

TUTORIAL INTEGRASI API DALAM KATALOG MUSIK ONLINE

**Aliffathur Muhammad Revan
Gilang Andhika Buwana
Syafrial Fachri Pane**



**PT. Penerbit Buku Pedia
2023**

TUTORIAL INTEGRASI API DALAM KATALOG MUSIK ONLINE

Penulis:

Aliffathur Muhammad Revan
Gilang Andhika Buwana
Syafrial Fachri Pane

ISBN:

Editor:

-

Penyunting:

-

Desain sampul dan Tata letak:

Gilang Andhika Buwana

Font:

Calibri

Penerbit:

PT. Penerbit Buku Pedia

Redaksi:

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

Distributor:

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

Cetakan Pertama, 2023

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apa pun tanpa ijin tertulis dari penerbit

KATA PENGANTAR

Segala puji bagi Allah, Tuhan Yang Maha Esa atas karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan buku ini. Salawat serta salam semoga senantiasa tercurahkan kepada Nabi Besar Muhammad SAW.

Kami ucapkan juga rasa terimakasih kami kepada pihak-pihak yang mendukung lancarnya buku ini mulai dari proses penulisan hingga proses cetak, yaitu orang tua kami, pembimbing kami, rekan-rekan kami kami, dan masih banyak lagi yang tidak bisa kami sebutkan satu per satu.

Adapun, buku kami yang membahas tentang Integrasi API Dalam Katalog Musik ini telah selesai kami buat secara semaksimal dan sebaik mungkin agar menjadi manfaat bagi pembaca yang membutuhkan bantuan dalam membangun sebuah aplikasi atau rancangan sistem.

Kami sadar, masih banyak luput dan kekeliruan yang tentu saja jauh dari sempurna tentang buku ini. Oleh sebab itu, kami mohon agar pembaca memberi kritik dan juga saran terhadap karya buku ajar ini agar kami dapat terus meningkatkan kualitas buku.

Demikian buku ajar ini kami buat, dengan harapan agar pembaca dapat memahami informasi dan juga mendapatkan wawasan mengenai bidang sistem informasi manajemen serta dapat bermanfaat bagi masyarakat dalam arti luas. Terima kasih.

Bandung, 15 Mei 2023

Penulis

PRAKATA

Dalam era digital yang terus berkembang, teknologi telah merasuki berbagai aspek kehidupan, termasuk dalam bidang hiburan. Musik menjadi bagian penting dalam kehidupan manusia, dan saat ini kita dapat dengan mudah mengakses musik secara online melalui internet.

Salah satu teknologi yang populer saat ini adalah katalog musik online. Katalog musik online sering ingin menyediakan akses ke layanan pihak ketiga, seperti platform streaming musik, informasi tentang artis, atau data terkait musik lainnya. Dalam hal ini, API memainkan peran penting dengan memungkinkan katalog musik online terhubung dengan layanan pihak ketiga dan mengintegrasikan fitur-fitur tersebut ke dalam platform mereka. Melalui penggunaan API, data dan komunikasi antara katalog musik dan layanan pihak ketiga dapat disinkronkan dengan terstruktur, menciptakan pengalaman pengguna yang lebih beragam dan kaya.

Tujuan dari penulisan buku ini adalah untuk memberikan pemahaman tentang cara merancang dan mengintegrasikan API pada katalog musik online. Setiap bab dalam buku ini disusun dengan langkah-langkah yang sesuai untuk pembaca yang masih pemula. Selain itu, setiap kode program yang dijelaskan dalam buku ini dapat diakses melalui link :

DAFTAR ISI

KATA PENGANTAR.....	i
PRAKATA.....	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABLE.....	vi
BAB 1 PENGENALAN TERHADAP API DALAM KATALOG MUSIK ONLINE... 1	
1.1 APA ITU API?	1
1.2 APA ITU KATALOG MUSIK ONLINE?	2
1.3 MENGAPA API DIPERLUKAN DALAM KATALOG MUSIK ONLINE?	2
BAB 2 METOLOGI PEMBUATAN APLIKASI..... 3	
2.1 BAHASA PEMROGRAMAN	3
2.1.1 FRONT-END.....	3
2.1.2 BACK-END	4
2.1.3 JAVASCRIPT	4
2.1.4 GOLANG	5
2.1.5 GIN	6
2.1.6 HTML.....	6
2.1.7 CSS	7
2.1.8 BOOTSTRAP	7
2.1.9 API DEEZER.....	8
2.2 APLIKASI YANG DIGUNAKAN	9
2.2.1 GOLAND	9
2.2.2 XAMPP	9

2.2.3	PHPMYADMIN	10
2.2.4	GIT BASH	11
2.3	PENYIMPANAN YANG DIGUNAKAN	12
2.3.1	MySQL.....	12
BAB 3	PERANCANGAN DAN PENGEMBANGAN SISTEM	13
3.1	PROSES BISNIS YANG AKAN DIBANGUN	13
3.2	PERANCANGAN SISTEM (UML)	14
3.3	USE CASE DIAGRAM.....	14
BAB 4	TUTORIAL PEMBUATAN APLIKASI	18
4.1	MEMBUAT DATABASE	18
4.2	MENGAMBIL API.....	24
4.3	MEMBUAT BACK-END.....	29
4.4	MEMBUAT FRONT-END	36
	DAFTAR PUSTAKA.....	45
	GLOSARIUM.....	47
	TENTANG PENULIS	57

DAFTAR GAMBAR

Gambar 1 Application Programming Interface	1
Gambar 2 Ilustrasi Front-end	3
Gambar 3 Ilustrasi Back-end	4
Gambar 4 Logo Javascript	4
Gambar 5 Logo Golang.....	5
Gambar 6 Logo Framework Gin	6
Gambar 7 Logo HTML.....	6
Gambar 8 Logo CSS	7
Gambar 9 Logo Bootstrap	7
Gambar 10 Logo Deezer.....	8
Gambar 11 Logo GoLand.....	9
Gambar 12 Logo XAMPP	9
Gambar 13 Logo phpMyAdmin	10
Gambar 14 Logo Git Bash.....	11
Gambar 15 Logo MySQL.....	12
Gambar 16 Proses Bisnis Penggunaan Aplikasi.....	13
Gambar 17 Use Case Katalog Musik Online	15
Gambar 18 Tampilan XAMPP Control Panel	19
Gambar 19 Mengaktifkan Apache dan MySQL pada XAMPP Control Panel	19
Gambar 20 Tampilan Localhost.....	20
Gambar 21 Tampilan phpMyAdmin.....	20
Gambar 22 Tombol New pada phpMyAdmin	21
Gambar 23 Membuat database	21
Gambar 24 Membuat table database	22
Gambar 25 Memasukkan attributes pada table	22
Gambar 26 Menambahkan Primary Key	23
Gambar 27 Tombol Save	23
Gambar 28 Tampilan akhir database	23
Gambar 29 Tampilan Situs Web Deezer for Developers.....	24
Gambar 30 Mengambil API Chart	25
Gambar 31 Salin Tautan API.....	25

Gambar 32 Salin Tautan API ke dalam Postman	25
Gambar 33 Tampilan data-data dari API.....	26
Gambar 34 Data ID.....	26
Gambar 35 Data title	27
Gambar 36 Data Artist Name	27
Gambar 37 Data Cover Album	27
Gambar 38 Data Preview	28
Gambar 39 Inialisasi projek baru	29
Gambar 40 Menentukan lokasi file	29
Gambar 41 Membuat file baru pada Project Browser.....	30
Gambar 42 Membuat file main.go.....	30
Gambar 43 Menginstall framework Gin.....	30
Gambar 44 Membuat variable dari data yang telah ditentukan	31
Gambar 45 Membuat variable untuk fungsi Like.....	31
Gambar 46 Menginstall Driver MySQL.....	31
Gambar 47 Membuat fungsi main	32
Gambar 48 Membuat koneksi dengan database	32
Gambar 49 Membuat Router	33
Gambar 50 Membuat Router untuk fungsi Like.....	34
Gambar 51 Membuat fungsi IncrementLikeCount	34
Gambar 52 Membuat fungsi getLikeCount	35
Gambar 53 Membuat Direktori untuk Front-end	36
Gambar 54 Memberikan nama direktori	36
Gambar 55 Membuat direktori HTML file.....	37
Gambar 56 Membuat file index.html.....	37
Gambar 57 Membuat header pada index.html	37
Gambar 58 Membuat direktori CSS	38
Gambar 59 Membuat file style.css.....	38
Gambar 60 Menambahkan kode css.....	39
Gambar 61 Membuat tampilan Widget Preview Player	39
Gambar 63 Kode style.css #1	40
Gambar 62 Kode style.css #2	40
Gambar 64 Kode style.css #3	41
Gambar 65 Kode style.css #4	41

Gambar 66 Kode style.css #5	41
Gambar 67 Kode Javascript #1	42
Gambar 68 Kode Javascript #2	43
Gambar 69 Tampilan Akhir Aplikasi	44

DAFTAR TABLE

Table 1 Definisi Aktor	15
Table 2 Definisi Use Case.....	16

BAB 1

Pengenalan Terhadap API dalam Katalog Musik Online

1.1 Apa itu API?

API merupakan singkatan dari *Application Programming Interface* yaitu kumpulan fungsi, perintah, protokol yang menjembatani komunikasi antara aplikasi yang satu dengan aplikasi yang lain (Gunawan et al., 2022).



Gambar 1 Application Programming Interface

API memiliki peran penting dalam mempermudah komunikasi dan integrasi antara berbagai aplikasi dan sistem. Penggunaan API di berbagai sektor industri membantu meningkatkan efisiensi operasional, meluaskan fungsionalitas aplikasi, dan memfasilitasi kolaborasi antara platform yang berbeda.

API dapat digunakan dalam berbagai industri, seperti teknologi dan perangkat lunak, *E-commerce*, perbankan, transportasi, media dan hiburan, layanan pelanggan, *internet of things* (IoT), kesehatan, pendidikan, dan masih banyak lagi. API dapat hadir dalam berbagai bentuk, termasuk API web yang berinteraksi melalui protokol HTTP dan yang terkait dengannya, API yang berorientasi pesan, serta API yang berbasis komponen seperti Java API. Dokumentasi API yang berkualitas dan penggunaan standar seperti *Representational State Transfer* (REST) atau *Simple Object Access Protocol* (SOAP) membantu menyederhanakan proses integrasi aplikasi bagi para pengembang dengan sistem lainnya.

1.2 APA ITU KATALOG MUSIK ONLINE?

Seperti yang kita ketahui saat ini, manusia tidak terlepas dari musik. Dimanapun kita berada pasti kita sering mendengarkannya. Musik dipahami sebagai cabang seni yang membahas dan menetapkan berbagai suara ke dalam pola-pola yang dapat dimengerti dan dipahami manusia. Musik adalah bunyi atau suara yang diorganisir secara serasi dalam satuan temporal (Tika Lestari, 2020).

Sedangkan katalog online adalah suatu daftar yang dibuat secara elektronik yang bisa diakses secara online berbasis internet (Firliana et al., 2019). Katalog online dapat ditemukan dalam berbagai industri, seperti e-commerce, perpustakaan digital, industri pariwisata, industri musik, dan banyak lagi. Tujuan utama dari katalog online adalah memudahkan pengguna untuk mencari, menavigasi, dan memilih produk, layanan, atau konten yang mereka inginkan.

Jadi, Katalog musik online adalah daftar yang berisi data terkait musik yang dibuat secara elektronik dan dapat diakses melalui internet secara online. Tujuannya untuk memberikan pengguna akses yang mudah dan cepat untuk menemukan, memutar, mengelola koleksi musik secara online.

1.3 MENGAPA API DIPERLUKAN DALAM KATALOG MUSIK ONLINE?

Katalog musik online sering kali ingin menyediakan akses ke layanan pihak ketiga, seperti penyedia *streaming* musik, informasi artis, atau data terkait musik lainnya. API memungkinkan katalog musik online untuk terkoneksi dengan layanan pihak ketiga dan menghubungkan fitur-fitur tersebut ke dalam platform mereka. Melalui API, data dan komunikasi antara katalog musik dan layanan pihak ketiga dapat diselaraskan dengan terstruktur, sehingga menciptakan pengalaman pengguna yang lebih kaya dan beragam.

Katalog musik online juga terus berkembang dengan penambahan lagu, album, dan artis baru. Dengan adanya API, data katalog musik dapat diperbarui secara otomatis dari sumber eksternal, seperti label rekaman atau penyedia alat data musik. API memungkinkan penyelarasan secara *real-time* atau terjadwal, sehingga katalog musik tetap *up-to-date* tanpa perlu pembaruan manual yang memakan waktu.

BAB 2

METOLOGI PEMBUATAN

APLIKASI

2.1 BAHASA PEMROGRAMAN

Dalam bidang pengembangan aplikasi, pemahaman terhadap bahasa pemrograman adalah salah satu aspek yang sangat penting untuk dikuasai. Dalam penerapan API dalam katalog music online, terdapat dua jenis bahasa pemrograman yang digunakan, yaitu Front-end dan Back-end. Kedua jenis bahasa ini memiliki peran yang berbeda namun saling melengkapi untuk memastikan kinerja aplikasi tersebut optimal. Front-end berfungsi untuk mengatur tampilan dan *interface* pengguna, sedangkan Back-end berfungsi untuk mengatur bagian *server* dalam aplikasi.

Untuk Front-end pada Integrasi API dalam katalog music online ini akan menggunakan bahasa Javascript dan juga menggunakan bootstrap yang merupakan *framework* CSS. Untuk Back-end akan menggunakan *framework* Gin yang menggunakan Bahasa Golang.

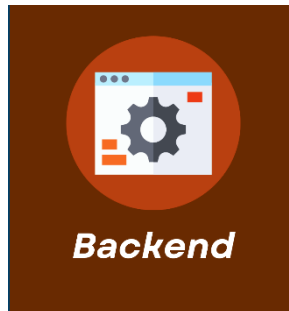
2.1.1 FRONT-END



Gambar 2 Ilustrasi Front-end

Front-End merupakan istilah yang mengacu pada *developer* yang bekerja di bagian desain *layout* suatu *website* atau aplikasi, dimana seorang *developer* akan mengerjakan user interface agar desain terlihat menarik dan tidak terlihat membosankan untuk dilihat. Seorang Front-End developer wajib menguasai program HTML, CSS, Javascript dan juga *framework* seperti jQuery, Bootstrap dan lain sebagainya (Widhyaestoeti et al., 2021).

2.1.2 BACK-END



Gambar 3 Ilustrasi Back-

Back-end dadalah tempat dimana proses pada suatu sistem informasi atau *aplikasi* berjalan, data dapat ditambahkan, diubah maupun dihapus. Back-end biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti *server* dan basis data. Back-end dibutuhkan dalam pengembangan sistem dan manajemen data pada sistem (Pangestika & Dirgahayu, 2020).

2.1.3 JAVASCRIPT



Gambar 4 Logo Javascript

Javascript adalah bahasa pemrograman yang awalnya dirancang untuk berjalan di atas *browser*. Namun, seiring perkembangan zaman, javascript tidak

hanya berjalan di atas *browser* saja. Javascript juga dapat digunakan pada sisi *Server, Game, IoT, Desktop*, dsb (petanikode.com & Muhardian, 2018).

JavaScript mengalami beberapa perubahan nama sepanjang waktu, dimulai dari Mocha, Mona, LiveScript, hingga akhirnya secara resmi mengadopsi nama JavaScript.

JavaScript secara alami terintegrasi dengan HTML dan CSS, yang merupakan dasar dari pembangunan web. Dengan JavaScript, Anda dapat memanipulasi elemen HTML, mengubah gaya CSS, dan mengubah struktur dan tampilan halaman web secara dinamis. Ini memberikan fleksibilitas dan kontrol yang lebih besar dalam membangun antarmuka pengguna yang kaya dan responsif.

2.1.4 GOLANG



Gambar 5 Logo Golang

Go Language atau biasa disebut Golang merupakan kepanjangan dari bahasa pemrograman satu ini. Seperti kita tahu namanya diawali dengan Go, ya benar dia merupakan bahasa pemrograman yang dikelola oleh Google. Google tidak bekerja sendirian, melainkan bekerja sama dengan 3 orang tokoh handal pada tahun 2009. Robert Griesemer, Rob Pike dan Ken Thompson merupakan ketiga tokoh tersebut (binaracademy.com, 2023).

Golang memiliki fungsi untuk pembuatan aplikasi server, pemrograman jaringan, pemrosesan data, pengembangan *aplikasi web*, dan pengembangan *aplikasi mobile*. Golang juga memiliki keunggulan diantaranya memiliki kinerja yang tinggi, kesederhanaan sintaksis, penanganan kesalahan yang baik, dan memiliki Pustaka standar yang kuat.

2.1.5 GIN



Gambar 6 Logo Framework Gin

Gin adalah salah satu jenis framework go yang paling populer dengan kinerjanya yang minimalis. *Framework* gin menggunakan *router* HTTP yang sangat cocok untuk pemula karena dokumentasinya yang sangat lengkap yang disediakan pada di Github (domainesia.com, 2020).

Gin memiliki beberapa kelebihan, seperti memberikan kinerja tinggi dengan menggunakan *routing* yang efisien dan pendekatan yang ringan, memiliki desain yang mudah dipahami, menyediakan sistem *middleware* yang kuat dan fleksible, memiliki dokumentasi yang lengkap dan mudah dipahami, serta memungkinkan pengembang untuk dengan mudah mengintegrasikan dengan pustaka dan paket eksternal lainnya yang dibutuhkan dalam pengembangan aplikasi.

2.1.6 HTML



Gambar 7 Logo HTML

Hypertext Markup Language (HTML) adalah standarisasi format file untuk transfer informasi via Internet melalui protokol HTTP, Protokol adalah aturan untuk melakukan komunikasi dalam sebuah jaringan, *port* yang digunakan untuk HTTP adalah 80 atau 8080 (Suryana et al., 2020).

Dengan menggunakan HTML dengan benar, pengembang dapat membangun halaman web yang terstruktur, menampilkan konten dengan jelas, mengatur tampilan dengan gaya yang sesuai, memberikan interaktivitas, dan meningkatkan pengalaman pengguna dalam mengakses halaman web.

2.1.7 CSS



Gambar 8
Logo CSS

Cascading Style Sheets (CSS) adalah standar teknologi pengembangan dalam pengaturan halaman web untuk menambahkan style seperti font, warna, jarak dan lainnya ke dokumen web (Rahmatika et al., 2020).

Dengan menggunakan CSS dengan baik, pengembang dapat mencapai tampilan dan tata letak yang konsisten, membuat halaman web responsif dan menarik, meningkatkan performa, serta memastikan kompatibilitas silang-*browser*. CSS merupakan alat yang penting dalam pengembangan web modern untuk mencapai desain yang menarik dan tampilan yang konsisten di berbagai perangkat.

2.1.8 BOOTSTRAP



Gambar 9 Logo
Bootstrap

Bootstrap adalah front-end framework yang bagus dan luar biasa yang mengedepankan tampilan untuk *mobile device* (Handphone, smartphone dll.) guna mempercepat dan mempermudah pengembangan *website*. Bootstrap menyediakan HTML, CSS dan Javascript siap pakai dan mudah untuk dikembangkan. Bootstrap merupakan *framework* untuk membangun desain web secara responsif. Artinya, tampilan web yang dibuat oleh bootstrap akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik di *desktop*, tablet ataupun *mobile device* (Suprayogi & Rahmanesa, 2019).

2.1.9 API DEEZER



Gambar 10 Logo Deezer

Deezer adalah layanan *streaming* musik *online* yang menyediakan akses ke jutaan lagu dari berbagai *genre* musik. Diperkenalkan pada tahun 2007, Deezer merupakan salah satu platform streaming musik terkemuka di dunia.

Deezer juga menyediakan layanan API untuk para *developer*. API Deezer mendukung berbagai operasi lain, termasuk streaming musik dalam aplikasi pengembang, mendapatkan informasi tentang stasiun radio, mencari berdasarkan rekomendasi musik, dan mengakses data trending musik. Pengembang yang menggunakan API Deezer harus mendaftar dan mendapatkan kunci API (API key) yang valid dari Deezer untuk mengotentikasi permintaan mereka. Deezer juga menyediakan dokumentasi yang komprehensif dan contoh kode untuk membantu pengembang memahami dan menggunakan API dengan baik.

2.2 APLIKASI YANG DIGUNAKAN

Sebagai seorang pengembang, penting untuk memiliki pengetahuan tentang alat dan perangkat yang digunakan dalam proses pengembangan sebelum memulai pengkodean dalam bahasa pemrograman. Dalam pembuatan Integrasi API dalam katalog musik, salah satu tool yang digunakan adalah GoLand sebagai *text editor* dan XAMPP sebagai software untuk menjalankan aplikasi pada *local server*. GoLand dipilih karena dalam pembuatan aplikasi ini menggunakan bahasa Golang.

2.2.1 GOLAND



Gambar 11 Logo GoLand

GoLand adalah lingkungan pengembangan terintegrasi (IDE) lintas *platform* untuk pengembangan bahasa Go. GoLand mencakup fitur-fitur seperti penyelesaian dan pemfaktoran ulang kode yang bergantung pada konteks, *debugging*, pembuatan profil, deklarasi navigasi, dan analisis kesalahan. Selain alat untuk pengembangan inti Go, GoLand juga mendukung untuk JavaScript, TypeScript, Node.js, SQL, Database, Docker, Kubernetes, dan Terraform. (Pranoto et al., 2022).

GoLand memiliki dukungan penuh untuk bahasa Golang, pemformatan kode otomatis yang sesuai dengan pedoman gaya resmi Go, refaktorisasi cerdas, dan penjelajahan kode yang mudah. Dalam keseluruhan, GoLand merupakan IDE yang kuat dan lengkap untuk pengembangan aplikasi Go, dengan banyak fitur yang dirancang khusus untuk meningkatkan produktivitas dan pengalaman pengembangan pengguna Go.

2.2.2 XAMPP



Gambar 12 Logo XAMPP

XAMPP adalah paket program web lengkap yang dapat dipakai untuk belajar pemrograman web, khususnya PHP dan MySQL. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP (Anggraini et al., 2020).

XAMPP menawarkan instalasi yang mudah dan cepat. Dalam satu paket, XAMPP menggabungkan semua komponen yang dibutuhkan untuk mengembangkan aplikasi web, termasuk server apache, *database* MySQL, *interpreter* PHP, dan bahasa pemrograman Perl. XAMPP tersedia untuk berbagai sistem operasi seperti Windows, macOS, dan Linux. Ini memungkinkan pengembang untuk mengembangkan aplikasi web secara konsisten di berbagai platform tanpa perlu mengubah konfigurasi atau lingkungan pengembangan.

2.2.3 PHPMYADMIN



Gambar 13 Logo phpMyAdmin

PhpMyAdmin adalah sebuah aplikasi atau perangkat berbasis *opensource* yang bisa kita gunakan secara gratis untuk melakukan pemrograman ataupun administrasi pada *database* MySQL. PhpMyAdmin sendiri menggunakan bahasa PHP untuk pemrogramannya (Ramadhan & Mukhaiyar, 2020).

PhpMyAdmin menyediakan antarmuka pengguna yang intuitif dan grafis untuk mengelola basis data MySQL. Pengguna dapat dengan mudah menjalankan perintah SQL, membuat, mengubah, dan menghapus tabel, mengimpor dan mengekspor data, mengelola pengguna dan hak akses, dan melakukan tugas-tugas pengelolaan database lainnya melalui antarmuka yang mudah digunakan. PhpMyAdmin dirancang untuk memudahkan pengguna dalam mengelola basis data MySQL. Tanpa perlu menggunakan baris perintah atau pengetahuan mendalam tentang SQL, pengguna dapat dengan mudah menjalankan tugas-tugas pengelolaan basis data melalui antarmuka web yang sederhana.

2.2.4 GIT BASH



Gambar 14 Logo Git Bash

Git Bash adalah sebuah terminal yang menyediakan lingkungan baris perintah (*command-line*) untuk pengguna Git. Git Bash dirancang untuk bekerja pada sistem operasi Windows dan memberikan antarmuka yang mirip dengan terminal Unix atau Linux. Dalam hal ini, Git Bash mengintegrasikan Git, yang merupakan sistem kontrol versi terdistribusi, dengan shell Unix yang disebut bash.

Dengan Git Bash, pengguna dapat menggunakan perintah-perintah Git dan menjalankan skrip shell di lingkungan Windows. Git Bash menyediakan sejumlah perintah yang serupa dengan perintah Unix, seperti `ls`, `cd`, `mkdir`, `rm`, dan sebagainya. Pengguna dapat menggunakan perintah-perintah ini untuk mengelola repositori Git, membuat *commit*, beralih ke cabang (*branch*), menggabungkan perubahan (*merge*), dan melakukan operasi-operasi lain yang terkait dengan Git.

2.3 PENYIMPANAN YANG DIGUNAKAN

Basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (Quispe, 2023).

Integrasi API dalam katalog musik online ini menggunakan *database* untuk menyediakan fitur like atau suka. Database yang digunakan dalam aplikasi ini yaitu menggunakan aplikasi MySQL.

2.3.1 MySQL



Gambar 15 Logo MySQL

MySQL adalah salah satu jenis database *server* yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses databasenya (Ramadhan & Mukhaiyar, 2020). MySQL telah ada dalam industri selama lebih dari dua dekade dan menjadi salah satu sistem manajemen basis data yang paling populer. Hal ini berarti ada banyak sumber daya, dokumentasi, tutorial, dan komunitas pengguna yang tersedia. Banyak organisasi dan pengembang memiliki pengalaman dan pengetahuan dalam menggunakan MySQL, sehingga mudah untuk mencari bantuan dan dukungan jika diperlukan.

MySQL dirancang untuk memberikan kinerja yang cepat dan efisien. Basis data ini dilengkapi dengan fitur-fitur seperti *caching*, indeks yang kuat, dan optimasi query, yang memungkinkan eksekusi query dan akses data yang cepat.

MySQL menyediakan fitur keamanan yang kuat untuk melindungi data sensitif. Ini termasuk pengelolaan pengguna dan hak akses yang fleksibel, enkripsi data, dan pemeriksaan otentikasi yang kuat. MySQL terus mengikuti perkembangan keamanan terbaru dan secara teratur merilis pembaruan keamanan untuk memastikan bahwa data tetap aman.

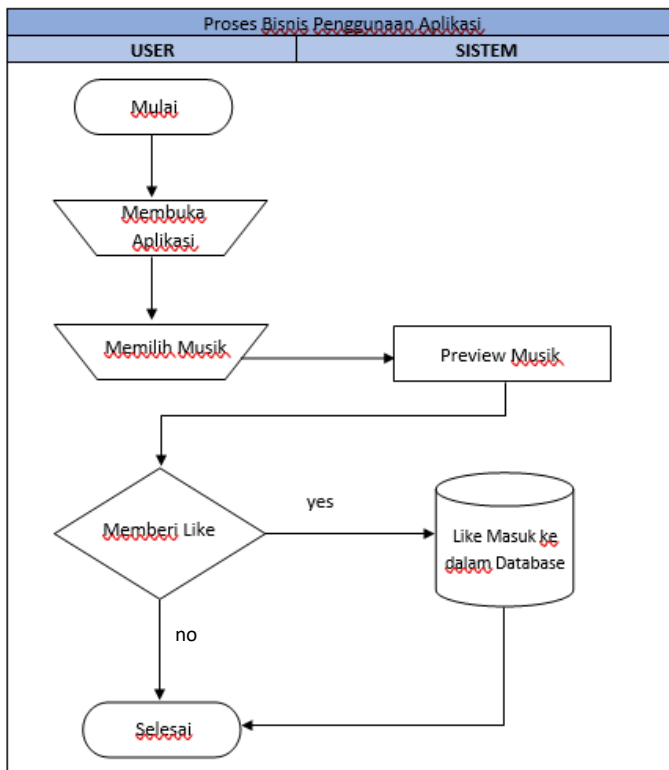
BAB 3

PERANCANGAN DAN PENGEMBANGAN SISTEM

3.1 PROSES BISNIS YANG AKAN DIBANGUN

1. Proses Bisnis Penggunaan Aplikasi

Pada alur ini, menjelaskan bagaimana user akan menggunakan aplikasi katalog music online ini. Proses Bisnis berikut merupakan gambaran alur proses penggunaan aplikasi oleh user.



Gambar 16 Proses Bisnis Penggunaan Aplikasi

Keterangan :

1. User membuka aplikasi.
2. User memilih musik yang sudah tersedia di dalam aplikasi.
3. Sistem akan memutar preview musik yang sudah dipilih oleh user sebelumnya.
4. User dapat menentukan apakah akan memberikan like pada preview musik yang sudah diputar atau tidak.
5. Jika user memberikan like pada preview musik , maka data like akan masuk ke dalam database.

3.2 PERANCANGAN SISTEM (UML)

UML adalah sebuah bahasa yang digunakan untuk menggambarkan sistem atau aplikasi yang berbasis objek. UML merupakan standar spesifikasi yang digunakan untuk mencatat, menjelaskan, dan membangun sistem perangkat lunak. UML juga berfungsi sebagai metodologi untuk membangun sistem berorientasi objek dan menyediakan alat-alat yang membantu dalam proses pengembangan. Dengan demikian, penggunaan UML dapat mempermudah pengembangan aplikasi yang berkelanjutan.

3.3 USE CASE DIAGRAM

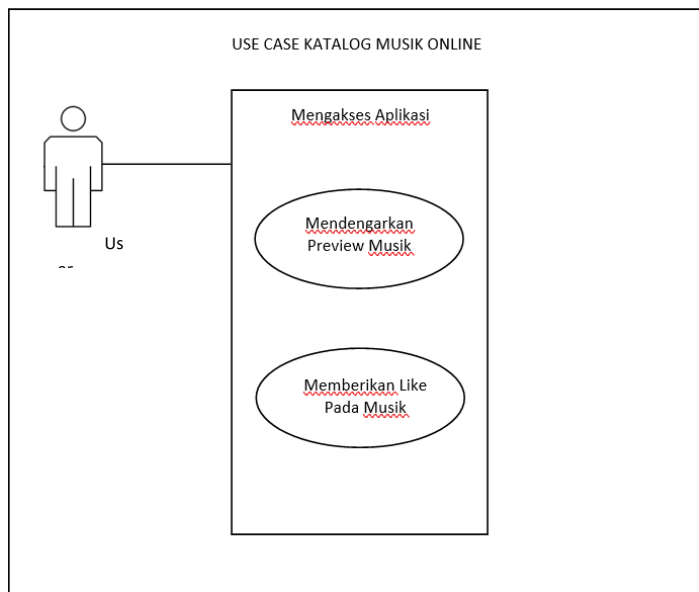
UML menyediakan cara penggunaan aplikasi melalui sebuah diagram yang dikenal sebagai Use Case Diagram. Bagian ini menggambarkan interaksi antara pengguna dan sistem dalam analisis sistem dan desain perangkat lunak. Diagram ini membantu dalam menggambarkan tugas atau aktivitas yang dapat dilakukan oleh sistem serta membantu dalam perencanaan dan spesifikasi sistem. Use Case Diagram menggambarkan sistem dalam bentuk lingkaran atau elips yang mewakili sistem itu sendiri, sedangkan pengguna diwakili oleh aktor. Use case sendiri diwakili oleh garis yang menghubungkan sistem dengan aktornya.

Diagram Use Case membantu dalam mengidentifikasi dan mengkategorikan use case yang ada dalam sistem. Ini adalah jenis diagram yang digunakan untuk menggambarkan interaksi antara sistem dan pengguna. Diagram ini membantu dalam menentukan batasan sistem dan kebutuhan sistem yang diperlukan selama proses analisis sistem dan desain perangkat lunak. Use case diagram sangat berperan penting dalam fase analisis sistem dan desain perangkat lunak karena memberikan pemahaman tentang bagaimana sistem akan digunakan oleh pengguna dan bagaimana sistem harus dikembangkan untuk memenuhi kebutuhan pengguna.

A. Use Case dan Definisi Aktor

aktor adalah entitas yang memiliki kepentingan dalam sistem dan melakukan interaksi dengan sistem untuk mencapai tujuannya. Aktor dalam use case dapat berupa individu, organisasi, atau sistem lain yang memiliki keterkaitan dengan sistem yang dijelaskan dalam use case.

Dalam sebuah use case, aktor dan sistem berinteraksi melalui serangkaian tindakan yang terkait dengan suatu tugas spesifik. Hasil dari interaksi ini adalah penyelesaian tugas tersebut dan pemenuhan kebutuhan aktor.



Gambar 17 Use Case Katalog Musik Online

Pada bagian ini akan dijelaskan definisi aktor yang terlibat dalam sistem.

Table 1 Definisi Aktor

No	Aktor	Deskripsi
1	User	<ul style="list-style-type: none">Mengakses AplikasiMendengarkan Preview Musik

		<ul style="list-style-type: none"> • Memberikan Like Pada Musik
--	--	--

Pada tabel 1. Defenisi aktor terdapat aktor dan deskripsi yang dilakukan berikut penjelasanya :

- a. User : Use case untuk User merupakan pengguna. Dalam use case ini, User dapat melakukan beberapa aktivitas yaitu Mengakses Aplikasi, Mendengarkan *Preview* Musik, dan Memberikan Like Pada Musik.
 - Mengakses Aplikasi : User dapat mengakses aplikasi tanpa harus melakukan registrasi atau *Login* terlebih dahulu.
 - Mendengarkan *Preview* Musik : Setelah masuk, User dapat mendengarkan *Preview* Musik yang sudah tersedia di dalam aplikasi.
 - Memberikan Like Pada Musik : Jika User menyukai preview musik yang sudah didengarkan, User dapat memberikan tanggapan berupa like.

B. Definisi Use Case

Table 2 Definisi Use Case

No	Aktor	Deskripsi
1	Mengakses Aplikasi	Merupakan aktivitas yang dilakukan oleh user. User dapat masuk ke dalam aplikasi tanpa harus melakukan registrasi ataupun <i>login</i>
2	Mendengarkan <i>Preview</i> Musik	Merupakan aktivitas yang dilakukan oleh user dimana user dapat mendengarkan potongan musik yang tersedia di dalam aplikasi.

3	Memberikan <i>Like</i> Pada Musik	Merupakan aktivitas yang dilakukan oleh User. Setelah User mendengarkan <i>preview</i> musik, user dapat memberikan tanggapan berupa like terhadap <i>preview</i> musik yang didengarkan.
---	-----------------------------------	---

BAB 4

TUTORIAL PEMBUATAN

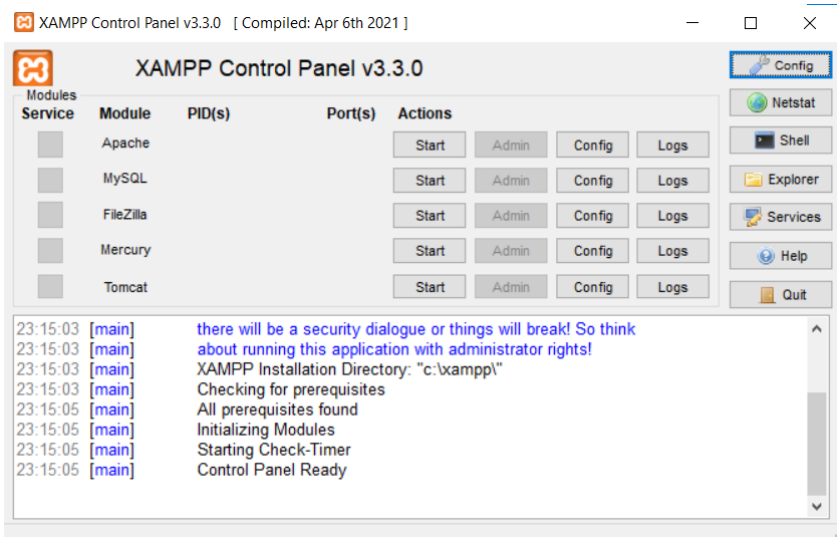
APLIKASI

4.1 MEMBUAT DATABASE

Untuk membuat aplikasi Integrasi API dalam katalog musik *online* ini kita membutuhkan sebuah database untuk menjalankan fitur like. Aplikasi ini tidak menggunakan banyak database dikarenakan aplikasi ini mengandalkan API dari *platform* Deezer.

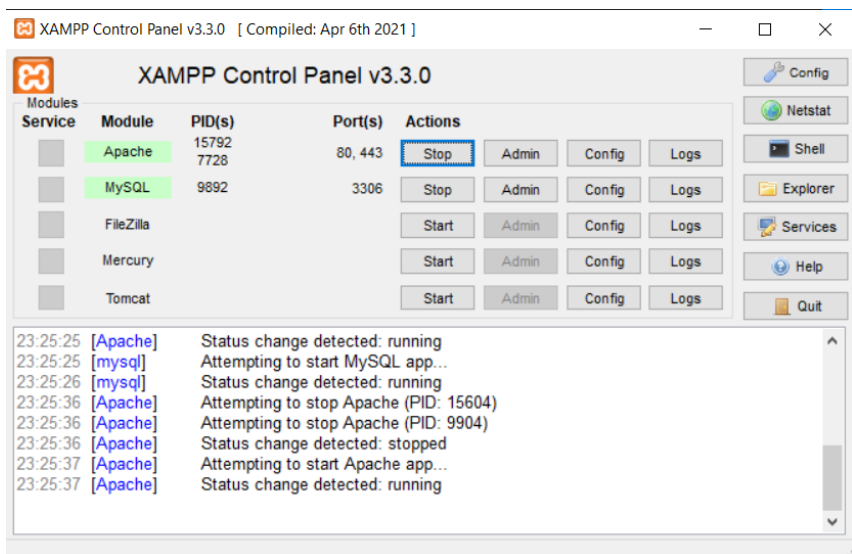
Berikut adalah Langkah-langkah membuat *database* untuk aplikasi Integrasi API dalam katalog musik *online* :

1. Anda harus mengunduh sebuah aplikasi bernama XAMPP yang mana merupakan paket *software* yang memungkinkan Anda untuk menjalankan aplikasi web lokal di komputer Anda. Anda dapat mengunduh XAMPP dari situs web resmi <https://www.apachefriends.org/download.html>
2. Setelah XAMPP terunduh, jalankan file instalasi kemudian Anda akan diminta untuk menyetujui lisensi perangkat lunak dan memilih lokasi instalasi. Pilih lokasi instalasi sesuai keinginan Anda kemudian tekan tombol "Next".
3. Setelah proses instalasi selesai, jalankan XAMPP Control Panel. Pada jendela yang muncul, Anda akan melihat beberapa layanan bawaan yang diaktifkan secara *default*.
4. Saat jendela XAMPP Control Panel terbuka maka akan muncul tampilan seperti berikut



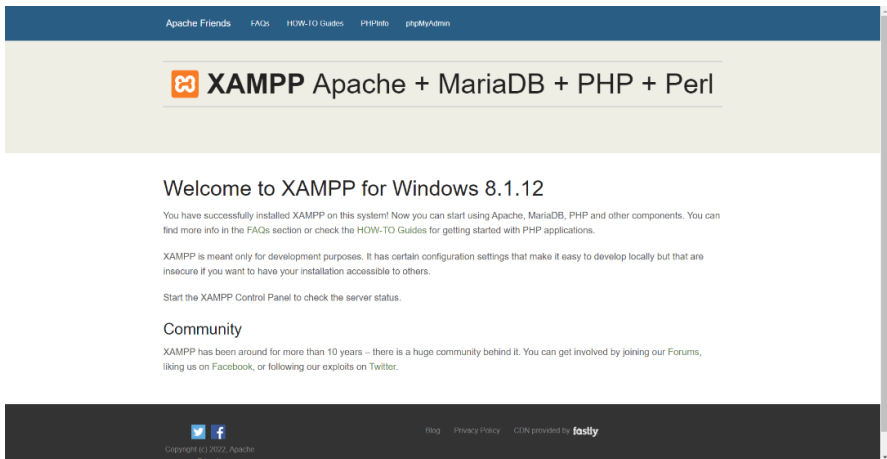
Gambar 18 Tampilan XAMPP Control Panel

- Aktifkan Apache dan MySQL pada XAMPP Control Panel. Tekan tombol "Start" pada kolom Apache dan MySQL. Tampilan XAMPP Control Panel akan menjadi seperti berikut



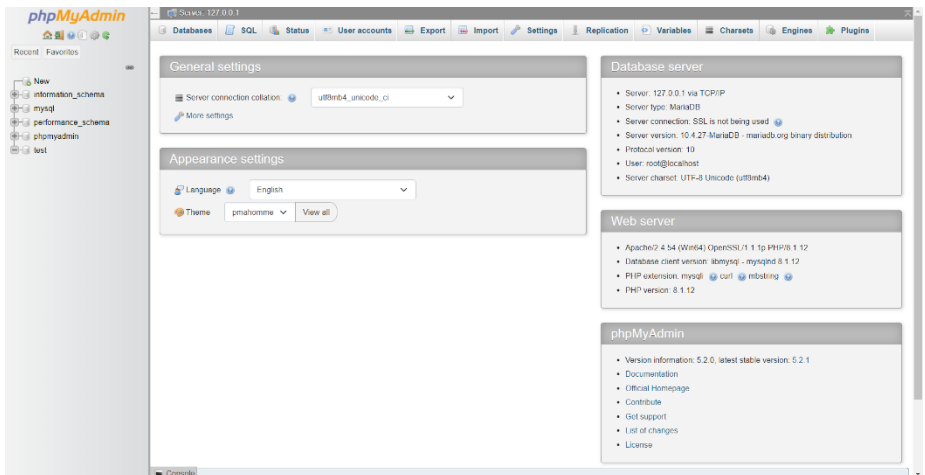
Gambar 19 Mengaktifkan Apache dan MySQL pada XAMPP Control Panel

6. Setelah itu, buka *browser* kemudian ketik “<https://localhost>” pada bar alamat dan akan muncul tampilan seperti berikut pada *browser*.



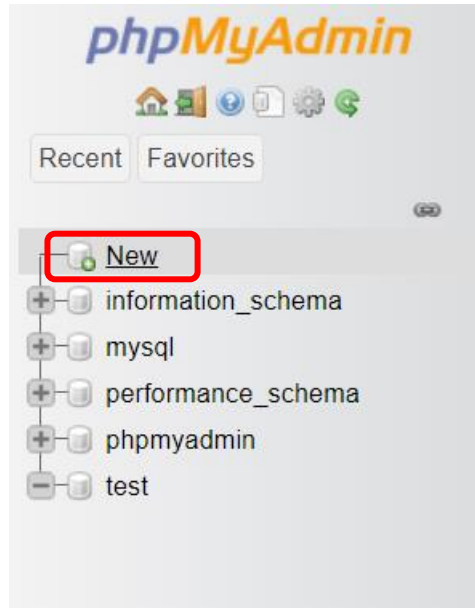
Gambar 20 Tampilan Localhost

7. Untuk mengelola *database*, Anda dapat menggunakan phpMyAdmin. Ketik “<http://localhost/phpmyadmin>” pada bar alamat di *browser* Anda atau klik tombol phpMyAdmin pada tampilan *localhost* Anda. Tombol phpMyAdmin berada di samping PHPInfo. Jika sudah maka akan muncul tampilan seperti berikut



Gambar 21 Tampilan phpMyAdmin

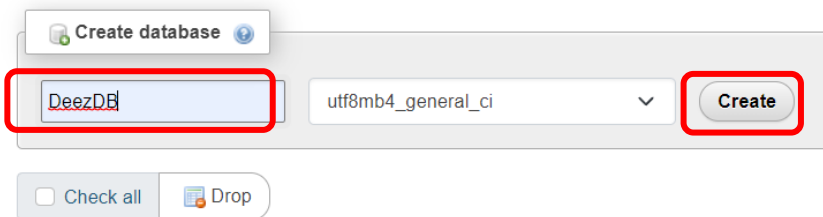
8. Klik “New” pada list *database* yang terdapat di bagian kiri pada tampilan phpMyAdmin.



Gambar 22 Tombol New pada
phpMyAdmin

9. Buat *database* dengan nama DeezDB sesuai rancangan pada BAB III kemudian tekan tombol “Create”.

Databases



Gambar 23 Membuat database

10. Buat table dengan nama “tracks” dan buat *Number of columns* sebanyak dua kolom kemudian tekan tombol “Create”.

Create new table

Table name

tracks

Number of columns

2

Create

Gambar 24 Membuat table database

11. Buat *attributes* pada table dengan nama `track_id` dan `like_counts`, keduanya memiliki tipe data *integer* atau *INT* dan buat `track_id` menjadi *primary key* pada kolom *Index*.

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Designer

Table name: tracks

Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
track_id	INT		None			<input type="checkbox"/>	PRIMARY
like_counts	INT		None			<input type="checkbox"/>	

Table comments:

Collation:

Storage Engine: InnoDB

Partition by:

(Expression or column list)

Partitions:

Preview SQL

Save

Gambar 25 Memasukkan attributes pada table

Add index
☐

Index name:

Index choice:

Advanced options

Column	Size
track_id [int]	

Preview SQL

Go Cancel

Gambar 26 Menambahkan Primary Key

- Setelah *primary key* telah dibuat kemudian tekan tombol “Save” yang terletak pada bagian bawah dan *database* pun selesai dibuat.

Preview SQL
Save

Gambar 27 Tombol Save

Server: 127.0.0.1 » Database: desozdb » Table: tracks									
Browse	Structure	SQL	Search	Insert	Export	Import	Privileges	Operations	Triggers
Table structure	Relation view								
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 track_id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2 like_counts	int(11)			No	None			Change Drop More
<input type="checkbox"/> Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext									
Print Propose table structure Move columns Normalize									

Gambar 28 Tampilan akhir database

4.2 MENGAMBIL API

Setelah kita membuat database, Langkah selanjutnya kita perlu mengambil API dari salah satu *platform* musik ternama yaitu Deezer. API dari Deezer dapat diambil dari situs web Deezer for developers.

Berikut adalah Langkah-langkah untuk mengambil API dari *platform* musik Deezer :

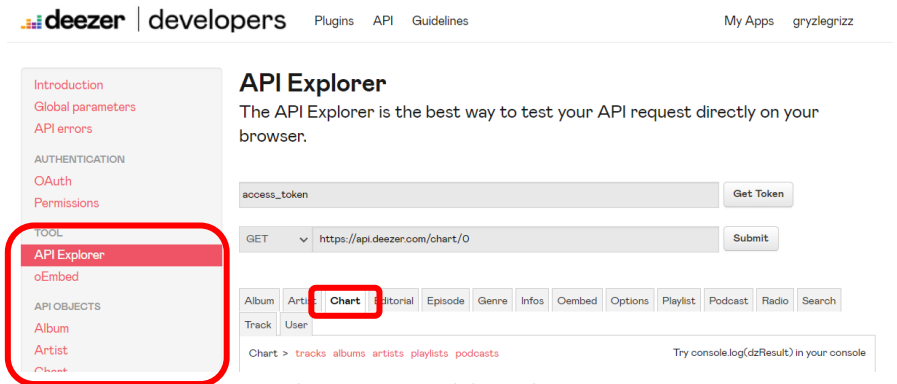
1. Buka halaman situs web resmi Deezer for Developers atau ketik “<https://www.deezer.com/api>” pada bar pencarian di *browser*. Jika sudah mengakses tautan tersebut maka akan terbuka dan tampilan situs web akan terlihat seperti ini

The screenshot displays the Deezer Developers API documentation page. The sidebar on the left contains a list of navigation links. The main content area is titled 'Introduction' and includes sections for 'Requests', 'Query quota', 'Requests format examples', and 'Response format'. The 'Requests' section shows an example URL: `https://api.deezer.com/version/service/id/method/?parameters`. The 'Query quota' section states that the number of requests per second is limited to 50 requests / 5 seconds. The 'Requests format examples' section shows three example URLs: `https://api.deezer.com/user/2529`, `https://api.deezer.com/user/2529/playlists`, and `https://api.deezer.com/album/302127`. The 'Response format' section shows an example GET request: `GET /search/artist/?q=eminem&index=&limit=&output=xml`. Below this, a table lists the available formats: JSON, application/json, and .json.

DESCRIPTION	ACCEPT HEADER	EXTENSION
JSON	application/json	.json

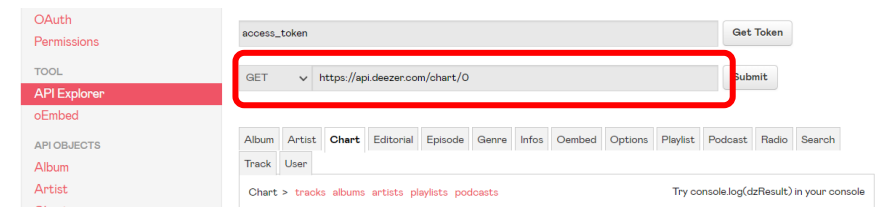
Gambar 29 Tampilan Situs Web Deezer for Developers

- Pilih “API Explorer” yang terdapat pada menu TOOL di sebelah kiri kemudian pilih “Chart”.



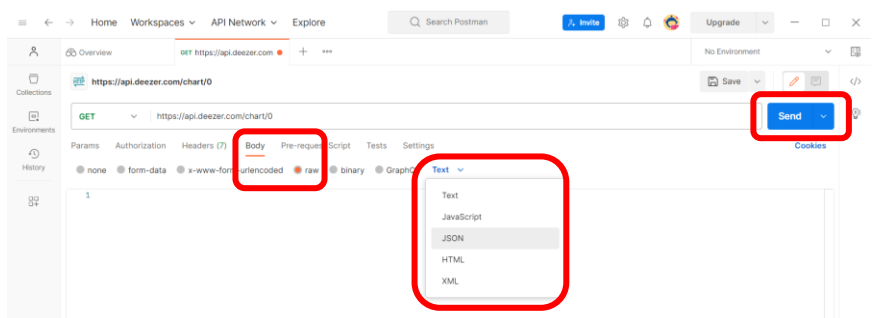
Gambar 30 Mengambil API Chart

- Salin tautan API seperti pada gambar berikut untuk nantinya dimasukkan ke Postman.



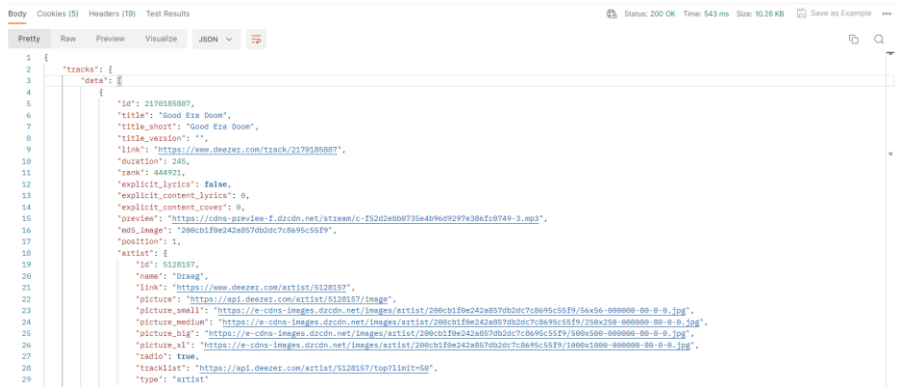
Gambar 31 Salin Tautan API

- Masuk ke postman kemudian tempel tautan yang sebelumnya telah disalin lalu pilih opsi “Body”, pilih “raw” dan ganti format “Text” menjadi format “JSON”. Setelah itu tekan tombol “Send”.



Gambar 32 Salin Tautan API ke dalam Postman

- Setelah itu maka akan muncul kumpulan data *tracks* dari API Deezer seperti ini



```
1 {
2   "tracks": [
3     {
4       "id": 2170185887,
5       "title": "Good Era Doom",
6       "title_short": "Good Era Doom",
7       "title_version": "",
8       "link": "https://www.deezer.com/track/2170185887",
9       "duration": 245,
10      "rank": 464921,
11      "explicit_lyrics": false,
12      "explicit_content_lyrics": 0,
13      "explicit_content_cover": 0,
14      "preview": "https://cdns-preview-f.dzcdn.net/stream/c-f52d2ebd8735e4b96d9297a386fc9749-3.mp3",
15      "md5_image": "298cb1f9e242a857b2dc7c8a95c55f9",
16      "position": 1,
17      "artist": {
18        "id": 5128157,
19        "name": "Dzaag",
20        "link": "https://www.deezer.com/artist/5128157",
21        "picture": "https://api.deezer.com/artist/5128157/image",
22        "picture_small": "https://e-cdns-images.dzcdn.net/images/artist/298cb1f9e242a857b2dc7c8a95c55f9/56x56-000000-00-0-0.jpg",
23        "picture_medium": "https://e-cdns-images.dzcdn.net/images/artist/298cb1f9e242a857b2dc7c8a95c55f9/250x250-000000-00-0-0.jpg",
24        "picture_big": "https://e-cdns-images.dzcdn.net/images/artist/298cb1f9e242a857b2dc7c8a95c55f9/500x500-000000-00-0-0.jpg",
25        "picture_xl": "https://e-cdns-images.dzcdn.net/images/artist/298cb1f9e242a857b2dc7c8a95c55f9/1000x1000-000000-00-0-0.jpg",
26        "radio": true,
27        "tracklist": "https://api.deezer.com/artist/5128157/top?limit=50",
28        "type": "artist"
29      }
30    }
31  ]
32 }
```

Gambar 33 Tampilan data-data dari API

- Pilih data-data yang akan digunakan, untuk aplikasi yang akan kita buat saat ini kita memerlukan data “id” sebagai primary key dari *tracks* yang akan ditampilkan.



```
"data": [
  {
    "id": 2170185887,
    "title": "Good Era Doom",
    "title_short": "Good Era Doom",
    "title_version": "",
    "link": "https://www.deezer.com/track/2170185887"
  }
]
```

Gambar 34 Data ID

- Lalu kita butuh data “title” untuk judul yang akan ditampilkan.

```

"data": [
  {
    "id": 2170185887,
    "title": "Good Era Doom",
    "title_short": "Good Era Doom",
    "title_version": "",
    "link": "https://www.deezer.com/track/2170185887"
  }
]

```

Gambar 35 Data title

8. Dari data "artist" kita membutuhkan nama *artist* yang ditampilkan ke dalam variable "name".

```

"artist": {
  "id": 5128157,
  "name": "Draag",
  "link": "https://www.deezer.com/artist/5128157",
  "picture": "https://api.deezer.com/artist/5128157/image"
}

```

Gambar 36 Data Artist Name

9. Dalam data "album" kita membutuhkan url yang mana terdapat pada data "cover_big" untuk cover yang akan ditampilkan pada *widget*. "cover_big" digunakan agar kita mendapatkan resolusi cover yang tinggi.

```

"album": {
  "id": 412497407,
  "title": "Dark Fire Heresy",
  "cover": "https://api.deezer.com/album/412497407/image",
  "cover_small": "https://e-cdns-images.dzcdn.net/images/cover/412497407/50x50.jpg",
  "cover_medium": "https://e-cdns-images.dzcdn.net/images/cover/412497407/250x250.jpg",
  "cover_big": "https://e-cdns-images.dzcdn.net/images/cover/412497407/1000x1000.jpg"
}

```

Gambar 37 Data Cover Album

10. Data “preview” diperlukan sebagai sumber *preview* musik yang akan diputar pada aplikasi.

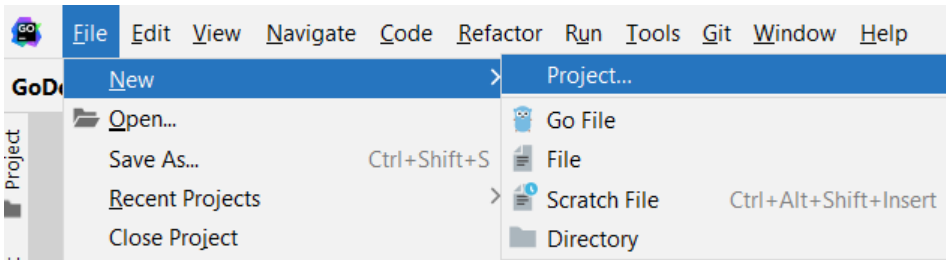
```
"data": [  
  {  
    "id": 2170185887,  
    "title": "Good Era Doom",  
    "title_short": "Good Era Doom",  
    "title_version": "",  
    "link": "https://www.deezer.com/track/2170185887",  
    "duration": 245,  
    "rank": 444921,  
    "explicit_lyrics": false,  
    "explicit_content_lyrics": 0,  
    "explicit_content_cover": 0,  
    "preview": "https://cdns-preview-f.dzcdn.net/stream/c-f5212el",  
    "md5_image": "200cb1f0e242a857db2dc7c8695c55f9",  
  }  
]
```

Gambar 38 Data Preview

4.3 MEMBUAT BACK-END

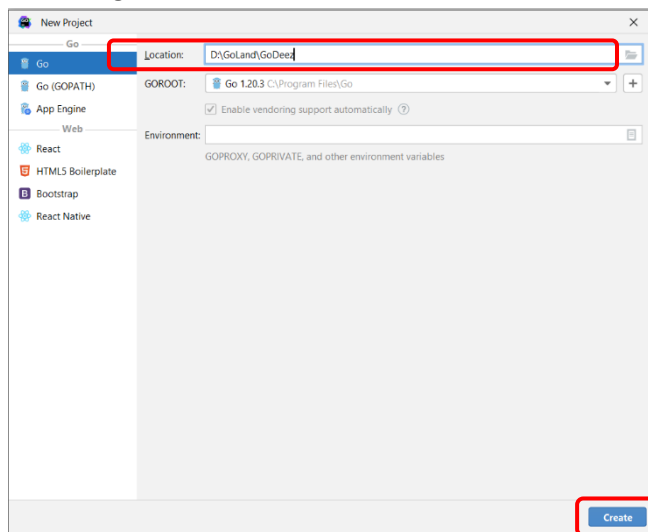
Setelah *database* dibuat dan data-data yang diperlukan dari API dikumpulkan, Langkah selanjutnya ialah membuat back-end. Disini kita menggunakan IDE GoLand, berikut adalah Langkah-langkah pembuatan Back-end menggunakan GoLand :

1. Buka GoLand dan inialisasi proyek baru dengan cara tekan tombol “File” pada bar *navigator* diatas lalu pilih “New” kemudian pilih “Project...”



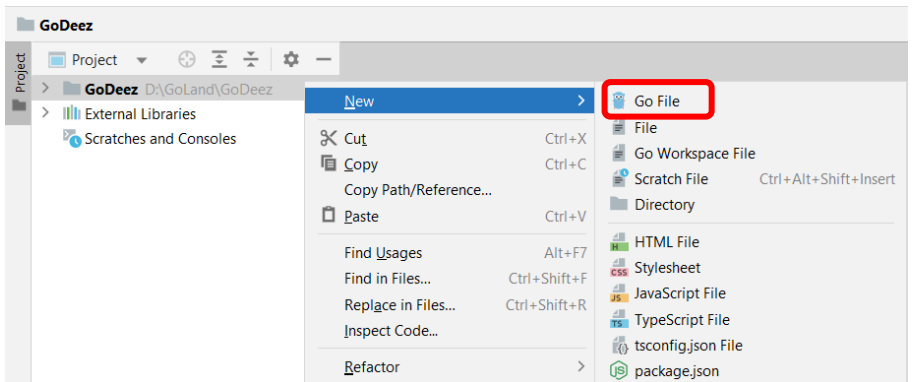
Gambar 39 Inisialisasi proyek baru

2. Akan muncul jendela untuk memilih lokasi pembuatan proyek. Pilih lokasi sesuai keinginan lalu tekan tombol “create”.



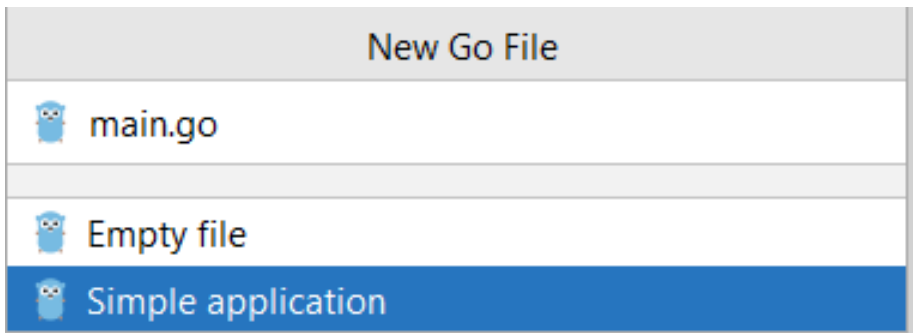
Gambar 40 Menentukan lokasi file

3. Buat file baru untuk membuat kode program utama yaitu “main.go” dengan cara klik kanan pada *Project Browser* lalu klik “New” dan pilih “Go file”.



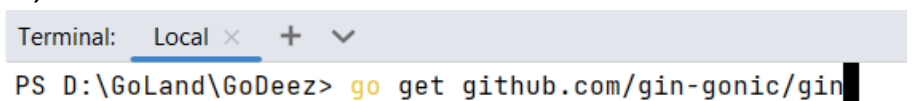
Gambar 41 Membuat file baru pada Project Browser

4. Buat nama file menjadi “main.go” kemudian pilih “Simple application” kemudian tekan tombol *Enter* pada *keyboard*.



Gambar 42 Membuat file main.go

5. Buka terminal yang berada di bawah kemudian *install framework gin* pada terminal dengan kode seperti berikut lalu tekan tombol *enter* pada *keyboard*.



Gambar 43 Menginstall framework Gin

6. Tunggu hingga instalasi selesai. Jika instalasi selesai sekarang masukkan data-data yang telah ditentukan tadi menjadi sebuah Variabel.

```

type Track struct {
    ID      int    `json:"id"`
    Title   string `json:"title"`
    Artist  struct {
        Name string `json:"name"`
    } `json:"artist"`
    Album  struct {
        CoverURL string `json:"cover_big"`
    } `json:"album"`
    PreviewURL string `json:"preview"`
}

```

Gambar 44 Membuat variable dari data yang telah ditentukan

7. Lalu buat variabel baru untuk fungsi *Like* seperti berikut ini.

```

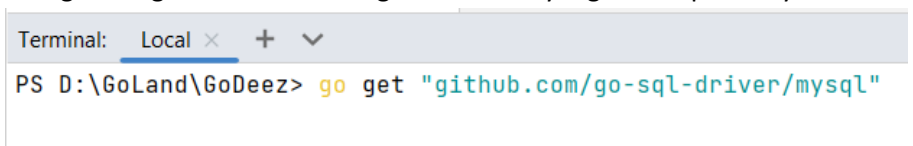
type TracksResponse struct {
    Data []Track `json:"data"`
}

type TrackWithLikeCount struct {
    Track      Track
    LikeCount  int
}

```

Gambar 45 Membuat variable untuk fungsi Like

8. *Install Driver MySQL* untuk Golang pada Terminal. Tujuannya untuk menghubungkan Back-end dengan *database* yang dibuat pada MySQL.



```

Terminal: Local x + v
PS D:\GoLand\GoDeez> go get "github.com/go-sql-driver/mysql"

```

Gambar 46 Menginstall Driver MySQL

9. Buat fungsi utama yang mendefinisikan lokasi *file* dan *directory static* dan *port* yang digunakan.



```

func main() {
    router := gin.Default()

    router.LoadHTMLGlob("view/*.html")
    router.Static("/css", "./view/css")

    log.Fatal(router.Run(":8080"))
}

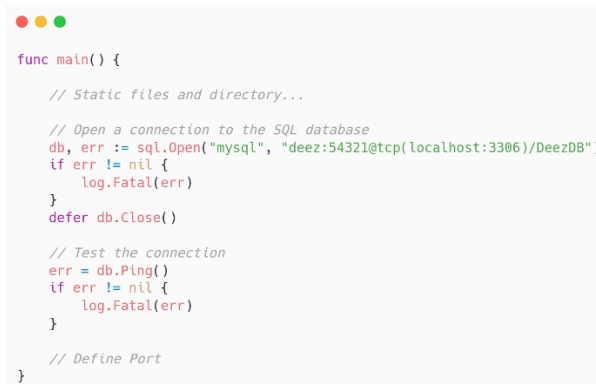
```

Static files and directory

Define

Gambar 47 Membuat fungsi main

10. Buat kode koneksi dengan *database* yang telah dibuat sebelumnya pada MySQL diantara kode *Static files and directory* dan *Define Port* .



```

func main() {

    // Static files and directory...

    // Open a connection to the SQL database
    db, err := sql.Open("mysql", "deez:54321@tcp(localhost:3306)/DeezDB")
    if err != nil {
        log.Fatal(err)
    }
    defer db.Close()

    // Test the connection
    err = db.Ping()
    if err != nil {
        log.Fatal(err)
    }

    // Define Port
}

```

Gambar 48 Membuat koneksi dengan database

11. Buat *router* baru untuk halaman utama dan tampilkan data dari API yang sudah ditampung dalam variabel. Kode *router* diletakkan setelah kode koneksi *database* dan sebelum kode *define port*. Router berfungsi untuk mengarahkan permintaan HTTP ke *handler* yang sesuai berdasarkan jalur permintaan (URL) dan metode HTTP.

```

func main() {

    // Static files and directory...
    // MySQL Database connection

    router.GET("/", func(c *gin.Context) {
        limit := 50 // Set the desired limit to retrieve more tracks, e.g., 50
        indexStr := c.Query("index")
        index, _ := strconv.Atoi(indexStr)

        // Construct the URL with pagination parameters
        url := fmt.Sprintf("https://api.deezer.com/chart/0/tracks?limit=%d&index=%d", limit, index)

        // Retrieve data from Deezer API
        response, err := http.Get(url)
        if err != nil {
            log.Fatal(err)
        }
        defer response.Body.Close()

        var tracksResponse TracksResponse
        err = json.NewDecoder(response.Body).Decode(&tracksResponse)
        if err != nil {
            log.Fatal(err)
        }

        // Get like counts for each track from the database
        tracksWithLikeCount := make([]TrackWithLikeCount, 0)
        for _, track := range tracksResponse.Data {
            likeCount := getLikeCount(track.ID, db)
            tracksWithLikeCount = append(tracksWithLikeCount, TrackWithLikeCount{
                Track: track,
                LikeCount: likeCount,
            })
        }

        // Render data in an HTML table
        c.HTML(http.StatusOK, "index.html", gin.H{
            "Tracks": tracksWithLikeCount,
        })
    })

    // Define Port
}

```

Gambar 49 Membuat Router

12. Buat *router* baru lagi untuk fungsi like pada aplikasi, buat kode ini setelah kode *router* halaman utama seperti berikut ini.

```
func main() {  
    // Main page router  
    router.POST("/like", func(c *gin.Context) {  
        var requestData struct {  
            TrackID int `json:"trackID"`  
        }  
  
        err := c.ShouldBindJSON(&requestData)  
        if err != nil {  
            c.JSON(http.StatusBadRequest, gin.H{"error": "Invalid request"})  
            return  
        }  
  
        // Increment the like count in the database  
        err = incrementLikeCount(requestData.TrackID, db)  
        if err != nil {  
            c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed to increment like count"})  
            return  
        }  
  
        c.JSON(http.StatusOK, gin.H{"success": true})  
    })  
    // Define Port  
}
```


Gambar 50 Membuat Router untuk fungsi Like

13. Buat fungsi “IncrementLikeCount” yang akan menambah jumlah *like* dan memperbarui *database* setiap kali user menekan tombol like.

```
func incrementLikeCount(trackID int, db *sql.DB) error {  
    // Check if the track exists in the database  
    query := "SELECT like_counts FROM tracks WHERE track_id = ?"  
    var currentLikeCount int  
    err := db.QueryRow(query, trackID).Scan(&currentLikeCount)  
    if err != nil {  
        if err == sql.ErrNoRows {  
            // Track does not exist, insert a new row  
            _, err = db.Exec("INSERT INTO tracks (track_id, like_counts) VALUES (?, 1)", trackID)  
            return err  
        }  
        return err  
    }  
  
    // Increment the like count  
    _, err = db.Exec("UPDATE tracks SET like_counts = ? WHERE track_id = ?", currentLikeCount+1, trackID)  
    return err  
}
```

Gambar 51 Membuat fungsi IncrementLikeCount

14. Terakhir, buatlah fungsi *getLikeCount* untuk mendapatkan jumlah *like* pada *database* dan ditampilkan pada tampilan aplikasi.



```
func getLikeCount(trackID int, db *sql.DB) int {
    query := "SELECT like_counts FROM tracks WHERE track_id = ?"
    var likeCount int
    err := db.QueryRow(query, trackID).Scan(&likeCount)
    if err != nil {
        if err == sql.ErrNoRows {
            // Track does not exist, return 0 like count
            return 0
        }
        log.Println("Failed to retrieve like count:", err)
        return 0
    }
    return likeCount
}
```

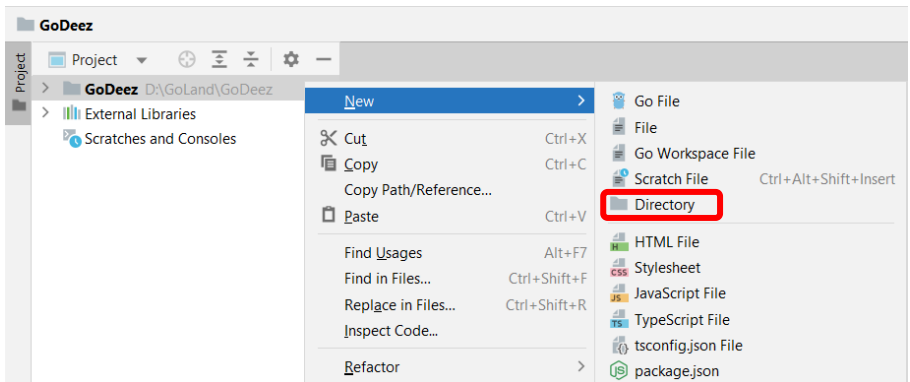
Gambar 52 Membuat fungsi getLikeCount

4.4 MEMBUAT FRONT-END

Setelah Back-end selesai dibuat maka langkah selanjutnya kita akan membuat Front-end atau tampilan agar data-data yang dibuat sebelumnya pada Back-end dapat ditampilkan dan agar aplikasi dapat diakses oleh pengguna.

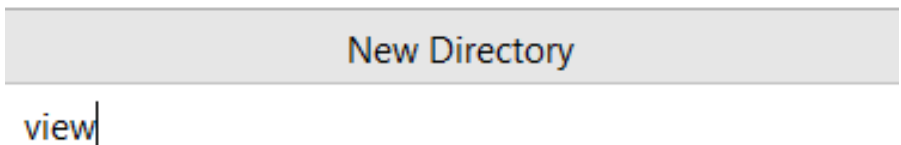
Berikut adalah langkah-langkah untuk membuat Front-end untuk aplikasi Integrasi API dalam katalog musik ini:

1. Langkah pertama yaitu buatlah direktori baru untuk menampung *file* Front-end dengan cara klik kanan pada Project *Browser* lalu klik “New” dan pilih “Directory”.



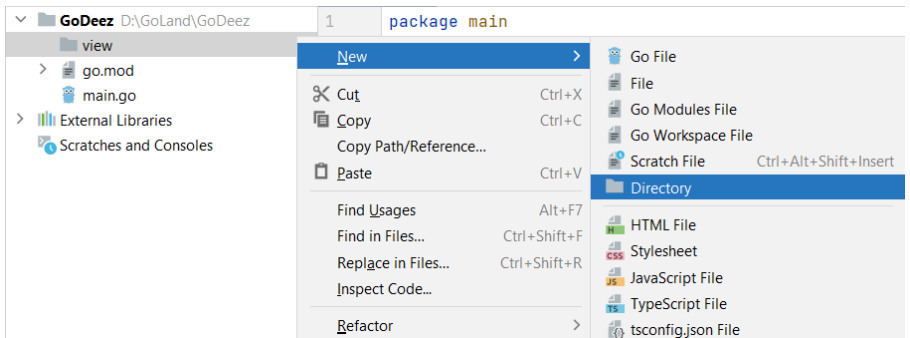
Gambar 53 Membuat Direktori untuk Front-end

2. Beri nama direktori sebagai “view”.



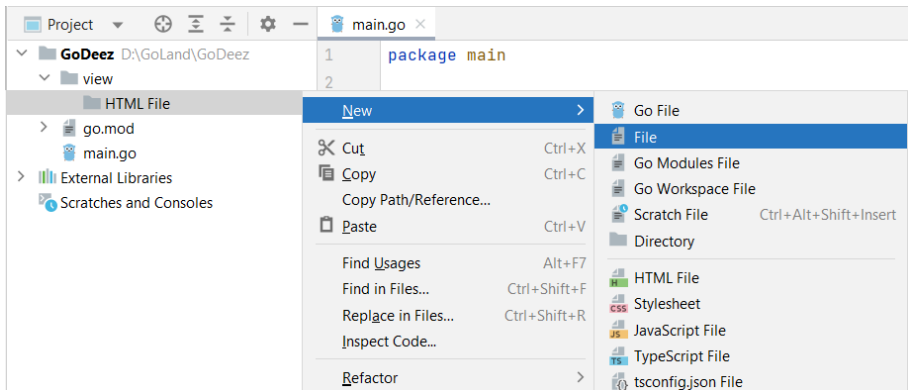
Gambar 54 Memberikan nama direktori

3. Buat direktori baru didalam direktori “view” dan beri nama direktori baru tersebut sebagai “HTML File”. Caranya yaitu klik kanan pada direktori “view” kemudian “New” dan pilih “Directory”.



Gambar 55 Membuat direktori HTML file

4. Buat file “index.html” pada direktori “HTML File”. Klik kanan pada direktori “HTML File” kemudian “New” dan pilih file, kemudian beri nama “index.html”.



Gambar 56 Membuat file index.html

5. Buat header pada file “index.html” seperti berikut

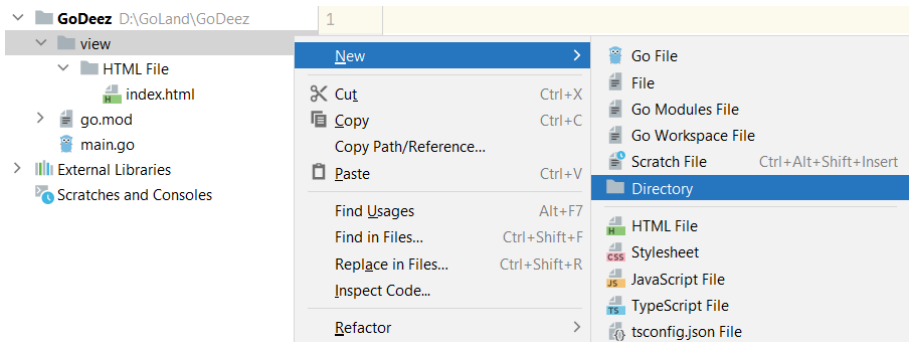
```

<html>
<head>
  <title>GoDeez</title>
  <link rel="stylesheet" href="css/main.css" />
  <!-- Icon Lib -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.3/font/bootstrap-icons.css">
</head>

```

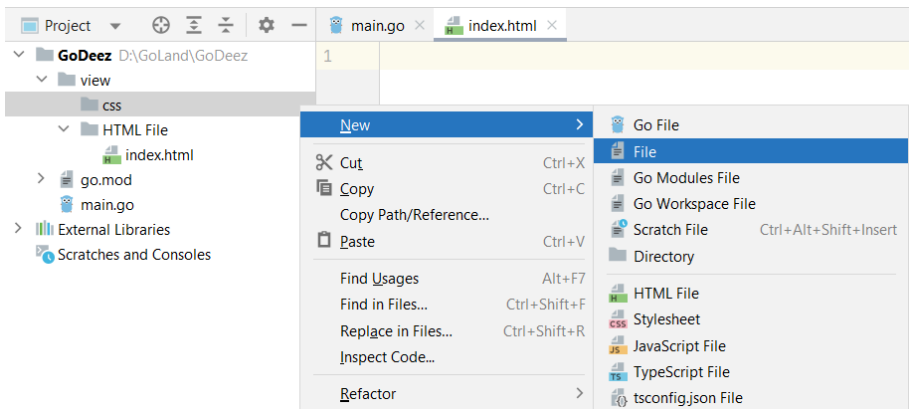
Gambar 57 Membuat header pada index.html

6. Buat direktori baru untuk menampung file css. Klik kanan pada direktori “view”, tekan “new” pilih “Directory” dan beri nama “css”



Gambar 58 Membuat direktori CSS

7. Buat file “style.css” pada direktori “css”. Klik kanan pada direktori “css” kemudian “New” dan pilih file, beri nama “style.css”



Gambar 59 Membuat file style.css

8. Buat kode seperti berikut untuk mengubah warna *background*, warna *font*, dan jenis *font* yang digunakan.

```

@import url('https://rsms.me/inter/inter.css');

html {
  font-family: 'Inter', sans-serif;
  scrollbar-width: none; /* For Firefox */
  -ms-overflow-style: none; /* For Internet Explorer and Edge */
}

html::-webkit-scrollbar {
  width: 0px; /* For Chrome, Safari, and Opera */
}

body {
  background-color: #0e0e0e;
  color: white;
}

```

Gambar 60 Menambahkan kode css

9. Buat tampilan *Widget Preview Player* pada file `index.html`

```

<body>
<div class="container">
  <h1>Made for you!</h1>
  <div class="grid-container">
    {{ range .Tracks }}
      <div class="grid-item">
        <div class="preview-card" style="background-image: url({{ .Track.Album.CoverURL }});"
        id="audioPlayer-{{ .Track.ID }}">
          <div class="content">
            <div class="playctl">
              <audio id="myAudio-{{ .Track.ID }}">
                <source src="{{{ .Track.PreviewURL }}" type="audio/mpeg">
              </audio>
              <div id="controls">
                <button class="ctrlBtn" class="playPauseButton" id="playPauseButton-{{ .Track.ID }}">
                  <i class="bi bi-play-fill"></i>
                </button>
                <div class="playerInfo">
                  <span class="trackTitleText">{{ .Track.Title }}</span>
                  <span class="trackArtistText">{{ .Track.Artist.Name }}</span>
                </div>
                <div class="progressBar-container2">
                  <div class="progressBar2" id="progressBar-{{ .Track.ID }}"></div>
                </div>
              </div>
            </div>
          </div>
          <div class="like-container">
            <button class="likeButton" id="likeButton-{{ .Track.ID }}" onclick="likeTrack({{ .Track.ID }})">
              <i class="fa fa-heart"></i>
            </button>
            <span id="likeCount-{{ .Track.ID }}">{{ .LikeCount }}</span>
          </div>
        </div>
      </div>
    {{ end }}
  </div>
</div>
</body>

```

Gambar 61 Membuat tampilan *Widget Preview Player*

Pada tampilan *Widget Preview Player*, kita menggunakan metode *Looping* dengan tujuan menampilkan *Widget* secara berulang

berdasarkan data *tracks* yang diambil dari API sebelumnya. Metode *Looping* diintegrasikan dengan kode `{{ range .Tracks }}` hingga `{{ end }}`.

10. Setelah itu, buat kode pada “style.css” sebagai berikut.

```
.container {
  margin-left: 12.5%;
  margin-right: 12.5%;
}

.preview-card {
  position: relative;
  border-radius: 2.5vh;
  width: 15vw;
  height: 15vw;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

.playctl {
  position: absolute;
  bottom: 0;
  left: 0;
  right: 0;
}
```

Gambar 63 Kode style.css #1

```
#audioPlayer {
  /* Style the audio player container */
}

#audioPlayer audio {
  display: none; /* Hide the default audio controls */
}

#audioPlayer #controls {
  /* Style the controls container */
}

#audioPlayer .playPauseButton {
  /* Style the play/pause button */
}

#audioPlayer .progressBar {
  position: sticky;
  bottom: 0;
  left: 0; width: 100%;
  height: 10px;
  background-color: #ccc; /* Set the color of the progress bar */
  margin-top: 10px;
  border-bottom-left-radius: 2.5vh;
  border-bottom-right-radius: 2.5vh;
}
```

Gambar 62 Kode style.css #2

```

.ctrlBtn {
  background-color: transparent;
  color: white;
  font-size: 2vw;
  border: none;
  text-align: left;
}

.playerInfo {
  padding: 1vw;
  display: flex;
  flex-direction: column;
}

.trackTitleText {
  font-weight: bold;
}

.trackArtistText {
  font-size: 2vh;
}

```

Gambar 64 Kode style.css #3

```

.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* Three equal width columns */
  grid-auto-rows: minmax(0%, 20vw); /* Minimum height of 100px for each row, expands as needed */
  grid-gap: 10px; /* Gap between grid items */
}

.grid-item {
  padding: 1%;
}

.progressBar-container2 {
  width: 100%;
  height: 1vw;
  background-color: #333333;
  border-radius: 2px;
  overflow: hidden;
  border-bottom-left-radius: 2.5vh;
  border-bottom-right-radius: 2.5vh;
}

```

Gambar 65 Kode style.css #4

```

.progressBar2 {
  height: 100%;
  background-color: white;
  width: 0;
  transition: width 0.2s ease;
}

.like-container {
  margin-top: 1vh;
}

.likeButton {
  background-color: transparent;
  color: white;
  font-size: 1.3vw;
  border: none;
  text-align: left;
}

```

Gambar 66 Kode style.css #5

11. Tuliskan Javascript yang menjadikan *widget* dapat berfungsi dan menampilkan waktu pada *progress Bar*. Javascript juga masuk kedalam

loop karena setiap *tracks* membutuhkan beberapa nama *Id*, fungsi, dan *class* berbeda. Oleh karena itu, memasukan *Script* ke dalam *loop* membuat kode menjadi lebih ringkas dan dinamis.

```
<script>
window.addEventListener("DOMContentLoaded", function() {
    var audios = document.querySelectorAll("[id^='myAudio']");
    var playPauseButtons = document.querySelectorAll("[id^='playPauseButton']");
    var progressBars = document.querySelectorAll("[id^='progressBar']");

    function handlePlayPauseButtonClick(index) {
        var audio = audios[index];
        var playPauseButton = playPauseButtons[index];

        if (audio.paused) {
            audio.play();
            playPauseButton.innerHTML = '<i class="bi bi-pause-fill"></i>'; // Change to pause icon
        } else {
            audio.pause();
            playPauseButton.innerHTML = '<i class="bi bi-play-fill"></i>'; // Change to play icon
        }
    }

    function handleTimeUpdate(index) {
        var audio = audios[index];
        var progressBar = progressBars[index];
        var progress = (audio.currentTime / audio.duration) * 100;
        progressBar.style.width = progress + "%";

        if (progress === 100) {
            progressBar.classList.remove("active");
        } else {
            progressBar.classList.add("active");
        }
    }

    for (var i = 0; i < playPauseButtons.length; i++) {
        (function(index) {
            playPauseButtons[index].addEventListener("click", function() {
                handlePlayPauseButtonClick(index);
            });

            audios[index].addEventListener("timeupdate", function() {
                handleTimeUpdate(index);
            });
        })(i);
    }
});
</script>
```

Gambar 67 Kode Javascript #1

```

<script>

// Fungsi Widget Player sebelumnya...

function likeTrack(trackID) {
    var likeCountElement = document.getElementById("likeCount-" + trackID);
    var likeCount = parseInt(likeCountElement.textContent);

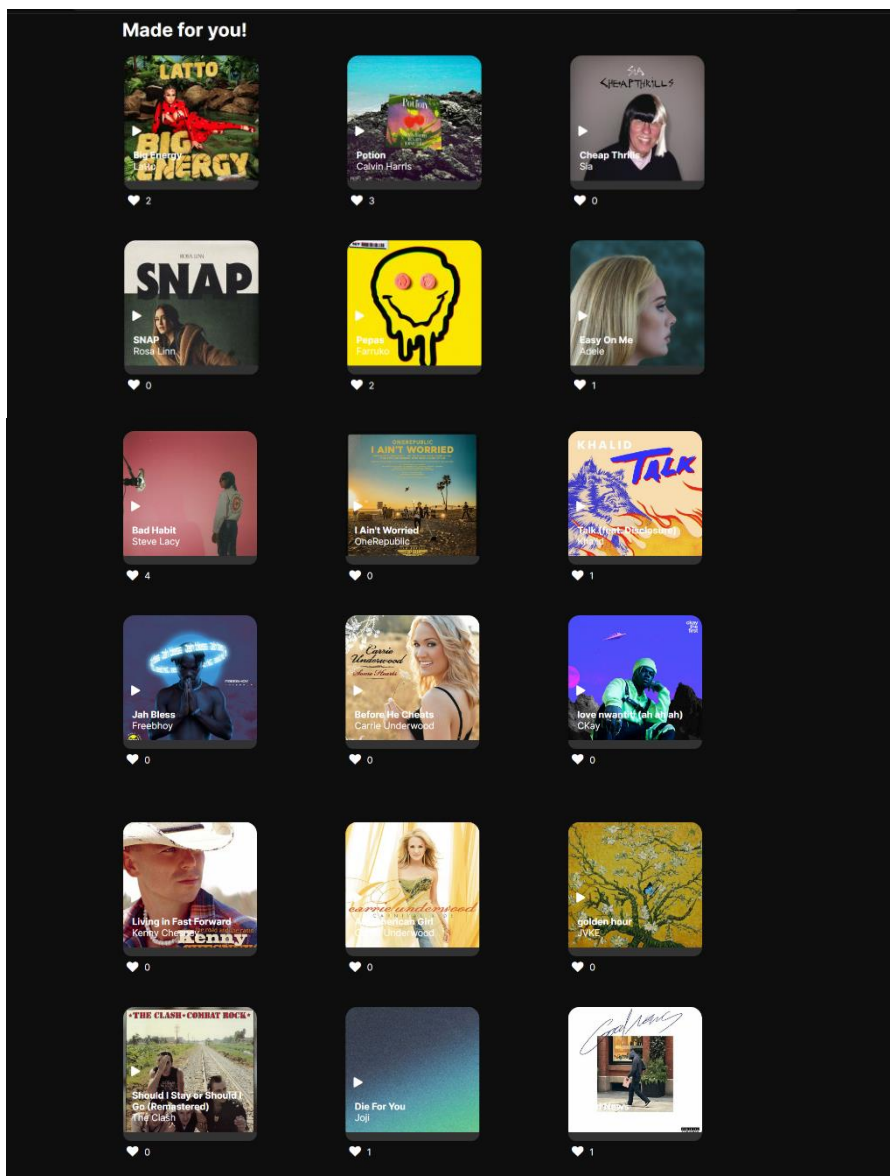
    // Send a POST request to increment the like count
    fetch("/like", {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify({
            trackID: trackID
        })
    })
        .then(function(response) {
            return response.json();
        })
        .then(function(data) {
            if (data.success) {
                // Increment the like count in the UI
                likeCount++;
                likeCountElement.textContent = likeCount;
            } else {
                console.log("Failed to increment like count");
            }
        })
        .catch(function(error) {
            console.log("Error:", error);
        });
}

</script>

```

Gambar 68 Kode Javascript #2

12. Jika semua tahapan sudah dilakukan maka aplikasi akan menampilkan musik-musik yang berada pada *top chart platform* musik Deezer dan user juga sudah bisa mendengarkan *preview* musik serta bisa memberikan tanggapan like pada musik-musik yang tersedia. Tampilan akhir aplikasi adalah seperti pada gambar berikut :



Gambar 69 Tampilan Akhir Aplikasi

DAFTAR PUSTAKA

Gunawan, F., Grace, D., & Bororing, M. G. (2022).

SISTEM PEMESANAN DAN PEMBAYARAN MAKANAN BERBASIS WEB TERINTEGRASI DENGAN APPLICATION PROGRAMMING INTERFACE (API).

Tika Lestari, D. (2020).

Membangun Harmoni Sosial Melalui Musik dalam Ekspresi Budaya Orang Basudara di Maluku.

Firliana, R., Indriati, R., Farida, I. N., & Niswatin, R. K. (2019).

APLIKASI E-KATALOG YUDISIUM BUKTI PENGAMBILAN IJAZAH DAN TRANSKRIP NILAI. In *Jurnal Inkofar* * (Vol. 1, Issue 2). Online.

Widhyaestoeti, D., Iqram, S., Mutiyah, S. N., & Khairunnisa, Y. (2021). Black Box Testing Equivalence Partitions Untuk Pengujian Front-End Pada Sistem Akademik Sitoda. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 7(3), 211–216.

Pangestika, R., & Dirgahayu, R. T. (2020).

Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta.

petanikode.com, & Muhardian, A. (2018).

BELAJAR PEMROGRAMAN JAVASCRIPT UNTUK PEMULA.
<https://www.petanikode.com/javascript-dasar/>

binaracademy.com. (2023).

Golang: Pengertian, Fungsi, dan Keunggulannya.
<https://www.binaracademy.com/blog/apa-itu-golang-dan-fungsinya>

domanesia.com. (2020).

Golang Framework: Pengertian, Kelebihan, Hingga Rekomendasi.
<https://www.domnesia.com/berita/golang-framework/#:~:text=Gin> adalah salah satu jenis, yang disediakan pada di Github.

Suryana, T., Kom, M., Amarullah, A., Donny, S. K., & Kom, R. S. (2020).

Pembelajaran HTML & DreamWeaver Pelatihan Pembuatan Website Dengan Menggunakan html dan Dreamwever Oleh: Pembelajaran Dasar.

Rahmatika, A. K., Pradana, F., & Abdurrahman Bachtiar, F. (2020).

Pengembangan Sistem Pembelajaran HTML dan CSS dengan Konsep Gamification berbasis Web (Vol. 4, Issue 8). <http://j-ptiik.ub.ac.id>

Suprayogi, B., & Rahmanesa, A. (2019).

PENERAPAN FRAMEWORK BOOTSTRAP DALAM SISTEM INFORMASI PENDIDIKAN SMA NEGERI 1 PACET CIANJUR JAWA BARAT (Vol. 6, Issue 2).

Pranoto, D., Handojo, A., & Satiabudhi, G. (2022).

Aplikasi War Game Pada Mobile Device Menggunakan Sensor Gyroscope dan Accelerometer. 10.

Anggraini, Y., Pasha, D., Damayanti, D., & Setiawan, A. (2020).

Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework Codeigniter. Jurnal Teknologi Dan Sistem Informasi, 1(2), 64–70. <https://doi.org/10.33365/jtsi.v1i2.236>

Ramadhan, R. F., & Mukhaiyar, R. (2020).

Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. JTEIN: Jurnal Teknik Elektro Indonesia, 1(2), 129–134. <https://doi.org/10.24036/jtein.v1i2.55>

Quispe, J. (2023).

No Titleการบริหารจัดการการบริการที่มีคุณภาพใน
โรงพยาบาลสังกัดกระทรวงสาธารณสุข. *วารสารวิชาการมหาวิทยาลัยอีสเทิร์นเอเชีย,*
4(1), 88–100.

GLOSARIUM

A

Aplikasi: Program yang dirancang untuk menjalankan tugas-tugas tertentu di komputer atau perangkat *mobile*.

Aplikasi *Mobile*: Aplikasi yang dikembangkan khusus untuk digunakan di perangkat *mobile*, seperti *smartphone* atau tablet.

Artist: Seseorang yang memiliki kemampuan atau bakat di bidang seni, seperti seniman lukis, aktor, atau penyanyi.

Attributes: Karakteristik atau sifat yang dimiliki oleh suatu objek dalam pemrograman.

B

Branch: Cabang atau percabangan dalam pengembangan perangkat lunak yang memungkinkan pengembang untuk bekerja pada versi yang berbeda secara paralel.

Browser: Program yang digunakan untuk mengakses dan menjelajahi halaman web di *internet*.

C

Caching: Proses menyimpan data sementara di memori komputer untuk mengaksesnya dengan lebih cepat di masa mendatang.

Chart: Grafik atau diagram yang digunakan untuk memvisualisasikan data atau informasi secara visual.

Class: Struktur atau templat yang digunakan untuk membuat objek dalam pemrograman berorientasi objek.

Commit: Tindakan menyimpan perubahan pada repositori dalam pengembangan perangkat lunak menggunakan sistem kontrol versi seperti Git.

D

Database: Kumpulan data yang terorganisir dan tersimpan secara sistematis untuk pengelolaan dan akses data.

Default: Nilai atau pengaturan bawaan yang akan digunakan jika tidak ada nilai atau pengaturan lain yang ditentukan.

Deklarasi: Pernyataan atau pengumuman dari variabel, konstanta, atau fungsi di pemrograman yang memberikan informasi tentang jenis data dan nama.

Desktop: Antarmuka pengguna grafis pada komputer yang berjalan di atas sistem operasi, biasanya berhubungan dengan pengaturan ikon dan *wallpaper*.

Developer: Seseorang yang terlibat dalam pengembangan perangkat lunak atau aplikasi.

Direktori: Tempat atau *folder* di sistem *file* yang digunakan untuk menyimpan dan mengatur berkas atau *file*.

Driver: Program yang memungkinkan perangkat keras atau perangkat lunak untuk berinteraksi dengan perangkat lainnya.

E

Eksekusi: Proses menjalankan atau menjalankan program atau instruksi dalam pemrograman.

Enkripsi: Proses mengubah atau mengkodekan informasi menjadi format yang tidak dapat dibaca atau dimengerti oleh pihak yang tidak berwenang.

External: Berkaitan dengan sesuatu yang berasal dari luar sistem atau perangkat yang sedang digunakan.

F

Framework: Kerangka kerja yang menyediakan struktur, aturan, dan alat untuk mengembangkan perangkat lunak.

Function: Blok kode yang dapat dipanggil untuk menjalankan tugas tertentu dalam pemrograman.

G

Game: Permainan yang dimainkan menggunakan perangkat elektronik, seperti komputer, konsol, atau ponsel.

Genre: Kategori atau jenis tertentu yang digunakan untuk mengklasifikasikan permainan, musik, film, atau karya seni lainnya.

H

Handler: Bagian dari program yang bertanggung jawab untuk menangani atau memproses suatu peristiwa atau permintaan.

Header: Bagian dari sebuah pesan atau dokumen yang berisi informasi tambahan, seperti judul, tanggal, atau informasi pengirim.

HTTP: Protokol komunikasi yang digunakan untuk mentransfer data melalui internet, umumnya digunakan untuk mengakses halaman web.

I

Identifikasi: Proses mengenali atau memastikan identitas seseorang atau sesuatu.

Instalasi: Proses memasang atau mengatur perangkat keras atau perangkat lunak agar siap digunakan.

Integrasi: Proses menggabungkan dua atau lebih sistem, komponen, atau aplikasi agar dapat bekerja bersama secara harmonis.

Interpreter: Program yang membaca dan menjalankan kode program secara langsung, baris per baris, dalam pemrograman.

IoT: *Internet of Things*; Konsep di mana objek-objek fisik seperti perangkat elektronik atau sensor terhubung ke *internet* dan saling berkomunikasi.

Intuitif: Mudah dipahami atau digunakan tanpa instruksi formal, berdasarkan naluri atau pemahaman alami.

J

K

Kompatibilitas: Kemampuan untuk bekerja atau berinteraksi secara harmonis dengan sistem atau perangkat lainnya.

Komprehensif: Mendalam atau mencakup semua aspek yang relevan, menyediakan pemahaman yang menyeluruh.

L

Layout: Tata letak atau susunan elemen dalam desain grafis, situs web, atau aplikasi.

Like: Tindakan mengekspresikan suka atau setuju terhadap sesuatu, umumnya diterapkan dalam media sosial atau platform online.

Lisensi: Izin legal atau hak yang diberikan untuk menggunakan, menggandakan, atau mendistribusikan suatu karya atau produk.

Localhost: alamat loopback IP pada komputer yang digunakan untuk mengakses layanan atau sumber daya yang berjalan pada mesin yang sama. Istilah localhost sering digunakan untuk merujuk pada server web lokal atau lingkungan pengembangan lokal.

Local server: Server yang beroperasi secara lokal di dalam jaringan komputer, biasanya digunakan untuk mengakses dan mengelola situs web atau aplikasi secara internal.

Login: Proses masuk ke dalam sistem atau aplikasi dengan menggunakan identitas atau kredensial yang valid, seperti nama pengguna dan kata sandi.

Looping: Pengulangan atau eksekusi berulang dari serangkaian pernyataan atau blok kode dalam pemrograman.

M

Metodologi: Pendekatan atau kerangka kerja yang digunakan dalam pengembangan perangkat lunak atau manajemen proyek.

Middleware: Perangkat lunak yang berfungsi sebagai perantara atau penghubung antara aplikasi atau sistem yang berbeda.

Mobile device: Perangkat elektronik yang dapat dibawa dan digunakan secara *mobile*, seperti *smartphone* atau tablet.

N

Navigator: Program atau perangkat lunak yang digunakan untuk menjelajahi atau mengelola halaman web di *internet*.

Navigasi: Proses bergerak atau menjelajah melalui halaman web, aplikasi, atau antarmuka pengguna.

O

Online: Terhubung ke internet atau beroperasi melalui jaringan.

Open source: Perangkat lunak yang kode sumbernya dapat diakses dan dimodifikasi oleh publik.

Optimasi: Proses meningkatkan kinerja, efisiensi, atau fungsionalitas suatu sistem, program, atau proses.

Orientasi: Penentuan atau pengaturan arah atau tata letak suatu objek atau elemen, terutama dalam desain grafis atau antarmuka pengguna.

P

Project: Proyek atau tugas yang memiliki tujuan tertentu, batas waktu, dan sumber daya yang dialokasikan untuk mencapai hasil yang diinginkan.

Platform: Lingkungan atau infrastruktur perangkat keras dan perangkat lunak yang digunakan untuk mengembangkan dan menjalankan aplikasi.

Player: Pengguna atau peserta yang menggunakan atau berinteraksi dengan suatu perangkat lunak, permainan, atau media.

Port: Pintu masuk atau saluran komunikasi yang digunakan untuk mentransfer data antara perangkat atau sistem yang berbeda.

Primary key: Kolom atau atribut dalam basis data yang secara unik mengidentifikasi setiap baris atau rekaman.

Preview: Pratinjau atau tampilan awal dari suatu konten atau desain sebelum disempurnakan atau dipublikasikan.

Protocol: Kumpulan aturan atau prosedur yang digunakan untuk mengatur komunikasi dan pertukaran data antara sistem atau perangkat.

Q

Query: Permintaan atau perintah yang digunakan untuk mengambil atau memanipulasi data dalam basis data.

R

Real-time: Keadaan atau kondisi di mana suatu proses atau data diperbarui atau ditampilkan secara langsung tanpa ada penundaan yang signifikan.

Registrasi: Proses pendaftaran atau pengisian formulir untuk menjadi anggota atau pengguna suatu layanan atau aplikasi.

Repository: Tempat penyimpanan atau koleksi yang berisi kode sumber, berkas, atau versi dari suatu proyek atau perangkat lunak.

Resolusi: Kualitas atau ketajaman gambar atau layar, atau proses memecahkan masalah atau mencari solusi dalam pemrograman.

Responsive: Kemampuan suatu desain, situs web, atau aplikasi untuk menyesuaikan tampilan dan fungsionalitas dengan perangkat atau ukuran layar yang berbeda.

REST: *Representational State Transfer*

Router HTTP: Perangkat keras atau perangkat lunak yang mengarahkan lalu lintas jaringan antara beberapa jaringan atau perangkat.

S

Script: Kode atau instruksi yang ditulis dalam bahasa pemrograman untuk melakukan tugas tertentu atau mengotomatisasi proses.

Server: Komputer atau sistem yang menyediakan layanan atau sumber daya kepada klien atau pengguna lain melalui jaringan.

SOAP: *Simple Object Access Protocol*; Protokol komunikasi yang digunakan untuk pertukaran data struktur dalam lingkungan distribusi objek.

Software: Program atau aplikasi yang dirancang untuk menjalankan tugas-tugas khusus di komputer atau perangkat elektronik.

Spesifikasi: Rincian atau persyaratan teknis yang harus dipenuhi dalam pengembangan atau implementasi suatu produk atau sistem.

Static: Tidak berubah atau tetap dalam pemrograman, biasanya digunakan untuk mengacu pada variabel atau metode yang tidak dapat diubah.

Streaming: Proses mengirim atau menerima data multimedia secara kontinu melalui jaringan untuk pemutaran segera.

Syntaxis: Aturan atau struktur tata bahasa yang digunakan dalam pemrograman atau penulisan kode.

T

Terminal emulator: Program atau aplikasi yang memungkinkan pengguna untuk mengakses dan menjalankan perintah melalui antarmuka teks di komputer.

Text Editor: Program atau aplikasi yang digunakan untuk menulis, mengedit, atau mengubah teks, biasanya digunakan dalam pemrograman atau penulisan.

Tracks: Jalur atau rekaman dalam audio atau video yang memisahkan atau mengorganisir konten dalam bentuk yang dapat dipisahkan.

U

Up-to-date: Dalam keadaan terkini atau dengan informasi terbaru yang relevan dan akurat.

URL: *Uniform Resource Locator*; alamat unik yang digunakan untuk mengidentifikasi sumber daya di internet, seperti halaman web atau berkas.

V

Variabel: Nama yang digunakan untuk menyimpan atau mereferensikan nilai atau data dalam pemrograman.

W

Website: Kumpulan halaman web yang terkait dan dapat diakses melalui internet.

Widget: Komponen atau alat kecil yang digunakan untuk menampilkan informasi atau fungsi tertentu dalam antarmuka pengguna.

X

Y

Z

TENTANG PENULIS



Gilang Andhika Buwana, lahir di Kota Bekasi, Jawa Barat pada tanggal 21 Mei 2004. Pendidikan tingkat dasar ditempuh di Kabupaten Purwakarta dan Kabupaten Bekasi serta pendidikan menengah ditempuh di Kabupaten Bekasi, tepatnya di Kecamatan Serang Baru. Saat buku ini ditulis, penulis sedang menempuh Pendidikan perguruan tinggi di Universitas Logistik & Bisnis Internasional.



Aliffathur Muhammad Revan, lahir di Kota Sibolga, Sumatera Utara pada tanggal 3 November 2003. Menempuh pendidikan tingkat dasar hingga menengah atas di Kota Bukittinggi, Sumatera Barat sebelum akhirnya melanjutkan Pendidikannya ke Universitas Logistik & Bisnis Internasional.



Syafrial Fachri Pane, S.T.,M.T.I.,EBDP sudah menyelesaikan pendidikan D3 di bidang Teknik Informatika di Politeknik Pos Indonesia (kini Universitas Logistik dan Bisnis Internasional), Ia juga menyelesaikan studi Sarjana di bidang Teknik Informatika di Universitas Pasundan, dan Magister di bidang Teknik Informatika di Universitas Bina Nusantara. Saat ini, ia sedang mengejar pendidikan doktoral (S3) di bidang Teknik Informatika di Telkom University.



JS

Buku ini berisi tentang cara mengintegrasikan API dengan langkah-langkah terperinci yang mencakup tahap pembuatan Back-end hingga aplikasi jadi

Tutorial ini dibuat bertujuan untuk membantu integrasi API dalam pembuatan aplikasi atau website dengan menggunakan Bahasa pemrograman Golang dan menggunakan database MySQL

= GO