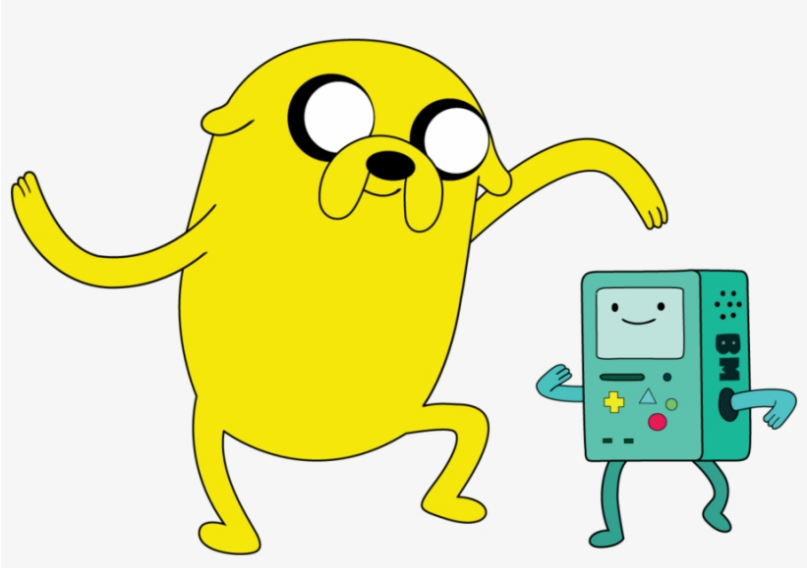
Spesifikasi Tugas Besar 1 IF3110

Milestone 1 - Monolithic PHP & Vanilla Web Application

Revisi 1: Pemindahan file ke teams. (18 Oktober 2022).
Revisi 2: Relasi mengenai penyanyi album dan lagu (26 Oktober 2022)

Deskripsi Persoalan



"BNMO dan Doni sedang berdansa mendengarkan musik toktok."

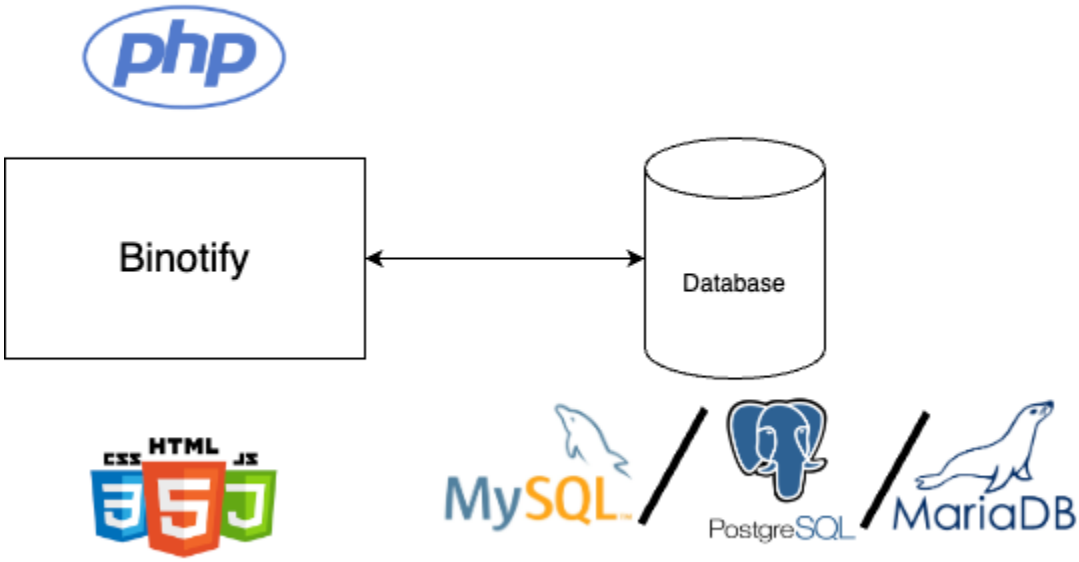
BNMO (dibaca: Binomo) dan Doni sedang tergila-gila dengan musik toktok yang berjudul "Tiba-Tiba". Musik tersebut dijalankan oleh BNMO dan Doni melalui aplikasi bernama *youcub* yang dinyalakan pada ponsel pintar milik Indra. Aplikasi *youcub* tersebut harus tetap dibuka agar lagu tetap berjalan dan Indra kesal karena ponselnya dipinjam terus menerus oleh BNMO dan Doni. Oleh karena itu, Indra memiliki ide untuk membuatkan **Binotify**, aplikasi musik berbasis web pada BNMO. Namun, permasalahannya, BNMO adalah mesin yang kuno sehingga hanya kuat untuk menjalankan sebuah DBMS (PostgreSQL/MariaDB/MySQL) dan PHP murni beserta HTML, CSS, dan Javascript vanilla. Karena Indra tidak paham mengenai hal tersebut, Indra meminta tolong mahasiswa Informatika angkatan 2020 untuk membuatkan aplikasi tersebut.

Tujuan Pembelajaran

- Mahasiswa mampu membuat sebuah aplikasi web dengan menggunakan HTML, CSS, dan JS.
- Mahasiswa mampu membuat sebuah layanan web dengan menggunakan PHP.
- Mahasiswa mampu menggunakan AJAX untuk melakukan pemanggilan yang bersifat *asynchronous*.
- Mahasiswa memahami penangananan state dan perbedaan HTTP method.

Spesifikasi Sistem

Gambaran Umum Sistem

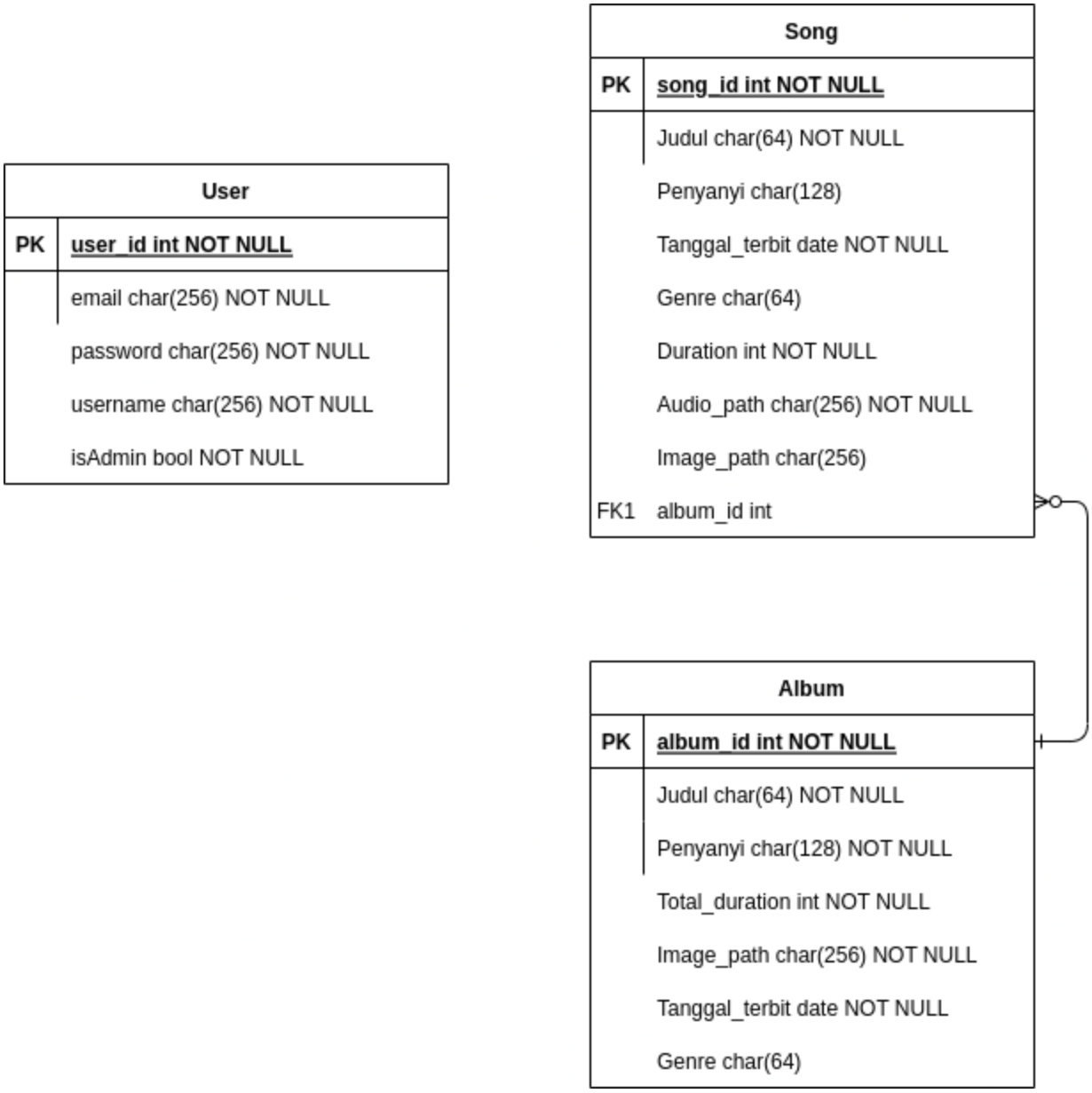


Gambaran Umum "Binotify" untuk Milestone 1.

Spesifikasi Umum

Berikut ada beberapa ketentuan umum yang harus dipenuhi oleh sistem.

1. Untuk *client-side*, gunakan **Javascript, HTML,** dan **CSS** secara murni.
   Tanpa menggunakan library atau **framework** CSS atau JS (e.g. JQuery, lodash, Bootstrap, atau tailwind). CSS harus berada di file yang berbeda dengan HTML (tidak inline styling, gunakan CSS selector).
2. Untuk *server-side*, wajib menggunakan **PHP murni** tanpa framework apapun (e.g laravel, codeigniter). Harap diperhatikan, Anda harus mengimplementasikan fitur menggunakan HTTP method yang tepat.
3. Untuk basis data, wajib menggunakan MySQL atau MariaDB atau PostgreSQL. Skema diharapkan mengikuti skema relasi yang sudah diberikan di bawah ini. Kalian bisa mengaplikasikan skema database dengan menambahkan tuning jika diperlukan. Penambahan/pengubahan dipersilakan selama tidak menghilangkan attribut yang tertulis pada skema di bawah ini.



4. (Opsional, hanya rekomendasi) Gunakan XAMPP untuk mempermudah pengerjaan.

User Story

1. Autentikasi
    a. Pengguna harus melakukan autentikasi untuk dapat mengakses seluruh fitur, **kecuali** apabila disebutkan fitur dapat diakses oleh pengguna yang tidak terautentikasi.
    b. Pengguna dibedakan menjadi 2 kategori: *user* dan *admin*.

    c. Seluruh pengguna yang terautentikasi juga harus dapat melakukan *logout*.
2. Admin dapat melakukan pengelolaan album dan musik.
3. Seluruh pengguna termasuk yang tidak terautentikasi dapat melihat daftar lagu dan memutar lagu.
4. Pemutaran lagu
    a. Pengguna yang tidak terautentikasi hanya dapat memutar lagu sebanyak 3 per hari. Pengguna tidak terautentikasi tidak boleh dibatasi 3 per hari secara global.
    b. Pengguna yang sudah terautentikasi dapat memutar lagu sebebasnya.
5. Seluruh pengguna termasuk yang tidak terautentikasi dapat melakukan search, filter dan sort pada daftar lagu berdasarkan judul atau penyanyi atau tahun terbit.
6. Seluruh pengguna termasuk yang tidak terautentikasi dapat melihat daftar lagu secara *pagination*.
7. Seluruh pengguna termasuk yang tidak terautentikasi dapat melihat daftar album dan detail album beserta list lagu pada album tersebut.
8. Terdapat *navigation bar* di setiap page yang ada.
9. Admin dapat melihat list user yang terdaftar.
10. Admin dapat melakukan pembuatan, penghapusan dan pengubahan lagu.
11. Admin dapat melakukan pembuatan, penghapusan dan pengubahan album lagu.

Pesan dari Asisten: “kalau kalian bertanya-tanya kenapa sistem autentikasinya kurang berguna. Dan cuman berfungsi supaya bisa dengar lagu sepuasnya, ditunggu milestone 2 😃, ini supaya nanti milestone 2 tidak banyak mengubah bagian milestone 1.”

Spesifikasi Fitur

Berikut ada beberapa ketentuan terkait fitur-fitur apa saja yang ada di dalam sistem.

1. **Autentikasi pengguna**

    Dalam melakukan autentikasi, pengguna perlu melakukan aksi login terlebih dahulu. Setelah sukses melakukan login, identitas pengguna akan disimpan dalam browser. Bentuk penyimpanan identitas pengguna dibebaskan kepada mahasiswa.

    - Untuk mengetahui pengguna mana yang sedang login, identitas ini dapat dicek di basis data.
    - Identitas ini tidak boleh disimpan sebagai parameter HTTP GET.
    - Jika identitas ini tidak ada, pengguna dianggap belum login dan diarahkan kembali ke halaman login dari halaman yang sedang dibukanya.
    - Ketentuan sebelumnya tidak berlaku untuk halaman yang dapat diakses tanpa autentikasi, yaitu Home, Detail Lagu, dan Detail Album.
    - Masa berlaku identitas pengguna dibebaskan.

2. Pengelolaan lagu dan album

   Admin dapat:

   - **Menambah** lagu
   - **Mengubah** detail lagu yang sudah ada.
   - **Menghapus** lagu.
   - **Melihat** detail lagu yang sudah ada.
   - **Mendengar** lagu.

   Pengguna biasa hanya bisa **mendengar** lagu.

   Ketika menambahkan lagu baru, perlu diperhatikan spesifikasi beberapa *field* berikut:

   - Tempat penyimpanan file lagu dibebaskan dimana saja (*local disk, cloud storage*, dll) asalkan tidak menyimpan *binary* file lagu di database. Database hanya menyimpan *path* menuju lokasi file.
   - Image, mekanisme penyimpanan sama seperti penyimpanan file lagu.
   - Durasi lagu dihitung otomatis oleh kode kalian ketika admin submit file lagu waktu menambah atau mengubah lagu. Jadi, di halaman tambah lagu atau ubah lagu, tidak ada *form* untuk input durasi lagu. Perhitungan durasi lagu dibebaskan di *client side* atau *server side*.

   Album adalah kumpulan lagu dengan penyanyi yang sama. Total durasi didapatkan dari akumulasi total semua durasi lagu yang ada di album. Admin dapat:

   - **Menambah** album baru.
   - **Menambah** dan **menghapus** daftar lagu yang ada di suatu album
   - **Menghapus** album.
3. Pencarian lagu

   Pengguna dan **Admin** dapat:

   - **Mencari** lagu berdasarkan **judul** atau **nama penyanyi** atau **tahun terbit**
   - **Menyortir** lagu berdasarkan tahun terbit secara *ascending* maupun *descending*.
   - **Filter** lagu berdasarkan genre

Spesifikasi Tiap Halaman

1. **Halaman Login**

   Halaman yang ditampilkan jika pengguna belum login atau sesudah logout adalah halaman Login. Pengguna dapat melakukan login sebagai *user* atau *admin*. Login hanya membandingkan username dan password saja. Tidak perlu tambahan proteksi apapun. Mekanisme autentikasi seperti yang dijelaskan sebelumnya.

Lakukan **hashing** pada password pengguna, algoritma yang digunakan dibebaskan.

2. **Halaman Register**

   Pengguna dapat mendaftarkan akun baru jika belum login atau sudah logout. Pada halaman ini, pengguna mendaftarkan diri pada sebuah form dengan *field* nama, *username* (unik), *email* (unik), *password*, dan *confirm password*. Pengguna tidak dapat mendaftar sebagai admin karena admin ditambahkan secara manual pada basis data. Pengecekan keunikan nilai field dilakukan menggunakan **AJAX**. Jika unik, border field akan berwarna hijau. Jika tidak unik, akan muncul pesan error pada form.

   Untuk mengurangi jumlah request yang dilakukan, mahasiswa disarankan untuk membuat sebuah fungsi ***debounce*** yang melakukan delay request setelah beberapa detik saat melakukan input.

   Validasi lain yang dilakukan pada sisi klien pada halaman ini adalah: *Email* memiliki format email standar seperti "example@example.com". *Username* hanya menerima kombinasi alphabet, angka, dan underscore.

   Setelah semua nilai *field* sudah diisi dan *valid*, pengguna dapat mendaftarkan akun barunya. Jika akun berhasil didaftarkan, pengguna langsung diarahkan ke halaman Home. Mekanisme autentikasi seperti yang dijelaskan sebelumnya.

3. **Halaman Home**

   Pada Halaman Home, ditampilkan username pengguna di kanan dan daftar lagu pada ruang yang tersisa.

   Daftar lagu ditampilkan secara **berurut** menurut **abjad** dari judul lagu, yakni dari a-z. Untuk setiap daftar lagu, ditampilkan informasi judul lagu, foto dari lagu, nama penyanyi lagu, tahun terbit lagu, dan genre lagu. Pengguna dapat melihat detail lagu dengan mengklik suatu lagu, kemudian diarahkan ke *Halaman Detail Lagu*. Tampilkan hanya 10 lagu terbaru (berdasarkan urutan di database).

   Header/navbar website dibedakan antara **user** dan **admin**.

   Untuk **user** terdiri dari search bar, daftar album, dan logout.

   Untuk **admin** terdiri dari menu tambah lagu, menu tambah album, daftar album, dan logout.

Search bar digunakan untuk mencari daftar lagu berdasarkan judul lagu, nama penyanyi, atau tahun terbit. Hasil pencarian ditampilkan pada *Halaman Search, Sort, and Filter*. Jika pengguna memilih untuk logout, pengguna akan diarahkan ke *Halaman Login*.

4. **Halaman Daftar Album**

   Untuk setiap daftar album, ditampilkan informasi judul album, nama penyanyi, tahun terbit album, dan genre album. Sama seperti *Halaman Home*, pengguna dapat melihat detail album dengan menekan bagian manapun pada section daftar album tersebut. Daftar album ditampilkan secara **berurut** menurut **abjad** dari judul album, yakni dari a-z.

   Disarankan mengaplikasikan juga pagination di page ini, dengan ketentuan pagination sama seperti *Halaman Search, Sort, and Filter*.

5. **Halaman Search, Sort, and Filter**

   Hasil pencarian dari search bar di *header/navbar website* akan ditampilkan pada halaman ini. Untuk setiap daftar lagu, ditampilkan informasi judul lagu, nama penyanyi lagu, tahun terbit lagu, dan genre lagu. Sama seperti *Halaman Home*, pengguna dapat melihat detail lagu dengan menekan bagian manapun pada section daftar lagu tersebut. Daftar lagu ditampilkan secara **berurut** menurut **abjad** dari judul lagu, yakni dari a-z.

   Pada halaman ini, pengguna juga dapat melakukan **sort** berdasarkan abjad dari judul lagu dan tahun terbit lagu, dan melakukan **filter** berdasarkan genre lagu.

   Jika daftar lagu melebihi jumlah tertentu (jumlah didefinisikan sendiri), maka akan muncul **pagination** untuk melihat daftar lagu selebihnya. Ketika memilih page, pengguna tidak diarahkan ke halaman baru, namun daftar lagu langsung berubah di halaman ini. **Pagination** wajib diimplementasikan pada **server side**.

6. **Halaman Detail Lagu**

   Halaman ini dapat diakses oleh pengguna dan admin. Data yang ditampilkan ketika pengguna mengakses halaman ini adalah:

   - Judul lagu
   - Penyanyi
   - Tanggal terbit
   - Genre
   - Durasi

- Tampilan yang dapat memainkan lagu (tombol play, tombol stop dan progress bar lagu).
- Cover photo.
- Tombol untuk melihat album dari lagu tersebut jika lagu tersebut bergabung dalam suatu album.

Tampilan halaman ketika admin mengakses halaman ini hampir sama dengan halaman pengguna. Bedanya adalah:

- Admin memiliki opsi untuk mengedit field yang ada kecuali durasi dan penyanyi.
- File uploader untuk mengubah file lagu serta cover photo.
- Tombol save untuk menyimpan perubahan.
- Admin juga memiliki opsi untuk menghapus lagu (perhatikan apa yang terjadi pada album jika ada lagunya yang dihapus).

7. **Halaman Detail Album**

   Halaman ini juga dapat diakses oleh pengguna dan admin. Data yang ditampilkan ketika pengguna mengakses halaman ini adalah:

   - Judul album
   - Penyanyi
   - Total durasi
   - Daftar lagu di album
   - Album cover

   Tampilan halaman ketika admin mengakses halaman ini hampir sama dengan halaman pengguna. Bedanya adalah:

   - Admin memiliki opsi untuk mengedit field kecuali total durasi (total durasi dihitung berdasar durasi tiap-tiap lagunya) dan penyanyi.
   - File uploader untuk mengganti album cover.
   - Tombol save untuk menyimpan perubahan.
   - Admin juga memiliki opsi untuk menghapus album.
   - Admin juga memiliki opsi untuk mengedit daftar lagu dari album terkait (perhatikan bahwa 1 lagu tidak bisa bergabung dalam 2 album yang berbeda).

8. **Halaman Tambah Album/Lagu**

   Tidak ada spesifikasi khusus untuk halaman ini. Silakan berkreasi sesuai dengan imajinasi kalian, asalkan memenuhi spesifikasi yang lain. Penambahan relasi antara album dan lagu **bisa** ditaruh di halaman ini.

9. **Halaman Daftar User**

   Halaman yang hanya dapat diakses oleh admin, menampilkan daftar user yang telah terdaftar. Informasi tiap user yang ditampilkan berupa username, email, dan informasi lainnya apabila ada field tambahan pada skema database user yang dirasa perlu ditampilkan.

Bonus

Catatan: Kerjakan dahulu spesifikasi wajib sebelum mengerjakan bonus.

1. **Responsive Web Design**

   Tampilan dibuat **responsif** (minimal untuk ukuran 1280 x 768 dan 800 x 600). Artinya, tampilan mungkin berubah menyesuaikan ukuran layar. Hint: gunakan CSS @media rule, lebih lanjut: https://www.w3schools.com/css/css_rwd_mediaqueries.asp.

2. **Docker**

   Membuat Dockerfile dari aplikasi kalian, serta membuat file docker-compose.yml dari aplikasi kalian yang berisi aplikasi kalian serta database yang digunakan. Tidak perlu membuat dockerfile yang kompleks, asalkan bisa dijalankan (kalian bisa saja membuat Dockerfile yang hanya berisi 3 baris). Tujuan bonus ini adalah agar kalian dapat mendapatkan pengalaman *hands-on* menggunakan docker.

3. **UI/UX seperti Spotify**

   Membuat UI/UX yang serupa dengan tampilan website Spotify, dengan desain yang konsisten dan responsif untuk setiap halaman. Siapa tahu nanti kalian bisa dapat pekerjaan di Spotify.

Keyword & Tips

Untuk meringankan beban tugas ini, ada beberapa *keyword* yang bisa Anda cari untuk menyelesaikan tugas ini di mesin pencari favorit kalian (selain *search engine* gatotkaca).

- **HTTP methods**: `GET, POST, PUT, DELETE`.
- **CSS**: `margin, padding, font-size, text-align, flex, grid, border, color, div, span, anchor tag, box-shadow`.
- **Javascript**: `XMLHTTPRequest, addEventListener`.
- **PHP**: `mysqli_connect, mysql_query, pg_connect, pg_query, PDO, $_GET, $_POST, $_COOKIE, var_dump, print_r, echo, require`.

- **SQL**: `SELECT, INSERT, UPDATE, DELETE, WHERE, LIKE, ORDER BY.`
- **DOCKER:** `Dockerfile, Docker Compose, PHP Dockerfile, Docker Container, Docker Image, Docker Volume.`

Berikut adalah *tips* dari asisten tercinta.

- Usahakan untuk membuat code yang maintainable (reusable, loosely coupled, dll). Sebagai contoh, dalam membuat navbar, jangan meng-copy navbar pada setiap file php namun manfaatkan **PHP templating**, lebih lanjut: https://css-tricks.com/php-templating-in-just-php/.
- Kalau mau eksperimen docker, pahami dulu istilah-istilah yang dipakai pada docker, jangan langsung mencoba, karena akan sulit apabila tidak berjalan atau menemukan *error*. Misalnya, pahami dulu perbedaan pembuatan *docker image* dan *docker container*. Lalu, ketahui cara kerja docker container mulai dari *port forwarding*, *docker volume*, dan *docker network*.
- Jangan berantakan buat saat buat kode *php*-nya. Ini saran agar keberlangsungan milestone 2 kalian lancar dan tidak terhambat oleh milestone 1 ini. Dan, apabila tugas besar ini dipakai untuk matkul lain, alur program dapat dibaca dan diingat dengan cepat. *(ini serius! ada matkul lain pakai tugas besar matkul ini)*

Responsi

Apabila keyword dan tips diatas masih sulit untuk dicerna. Akan diadakan responsi per kelompok bersama asisten menjelang pengumpulan tugas (tanggalnya menyusul, kemungkinan di hari Sabtu setelah UTS secara online atau seminggu menjelang deadline).

- Responsi bersifat **opsional**. Artinya **tidak wajib** dan jika mau saja.
- Kelompok yang ingin mendaftarkan diri dapat mengisi pada ~~sheet~~ ~~ini~~ teams.
- Perlu diperhatikan bahwa ketersediaan asisten **terbatas**.
  Siapkan terlebih dahulu hal-hal yang ingin ditanyakan / dikonsultasikan dan lakukan setup project di komputer anda **sebelum** sesi responsi dimulai.
- Pastikan sudah riset dan **mencoba** mempelajari sebelum bertanya ke asisten. Agar bisa bertanya lebih banyak pertanyaan dan pertanyaan lebih dekat ke pengerjaan bukan *understanding*.

Daftar Pertanyaan

Jika masih ada pertanyaan mengenai spesifikasi tugas, dapat dilakukan melalui ~~sheet~~ ~~ini~~ teams.

Pastikan tidak ada pertanyaan yang berulang.

Lain-Lain

Deliverables

Berikut adalah hal yang harus diperhatikan untuk pengumpulan tugas ini:

1. Buatlah grup pada Gitlab dengan format "IF3110-2022-KXX-01-YY", dengan XX adalah nomor kelas dan YY adalah nomor kelompok.
2. Tambahkan anggota tim pada grup anda.
3. **Fork** pada repository ini dengan organisasi yang telah dibuat.
4. Ubah hak akses repository hasil Fork anda menjadi **private**.
5. Hal-hal yang harus diperhatikan.
    a. Silakan commit pada repository anda (hasil fork).
    b. Lakukan beberapa commit dengan pesan yang bermakna, contoh: “add register form”, “fix logout bug”, jangan seperti “final”, “benerin dikit”, “fix bug”.
    c. Disarankan untuk tidak melakukan commit dengan perubahan yang besar karena akan mempengaruhi penilaian (contoh: hanya melakukan satu *commit* kemudian dikumpulkan).
    d. Sebaiknya *commit* dilakukan setiap ada penambahan fitur.
    e. *Commit* dari setiap anggota tim akan mempengaruhi penilaian.
    Jadi, setiap anggota tim harus melakukan commit yang berpengaruh terhadap proses pembuatan aplikasi.
    f. Sebagai panduan bisa mengikuti semantic commit.
6. Buatlah file **README** yang berisi:
    a. Deskripsi aplikasi web
    b. Daftar *requirement*
    c. Cara instalasi
    d. Cara menjalankan *server*
    e. *Screenshot* tampilan aplikasi (tidak perlu semua kasus, minimal 1 per halaman), dan
    f. Penjelasan mengenai pembagian tugas masing-masing anggota (lihat formatnya pada bagian pembagian tugas).

Pengumpulan Tugas

- Deadline tugas adalah pada hari **Jumat, 28 Oktober 2022 pukul 15.00 WIB**.
- Step pengumpulan adalah sebagai berikut.

    a. Buat release dengan format **vX** pada gitlab informatika dan judul bebas namun bermakna.
       X adalah adalah nomor yang dimulai dari 1.
    b. Apabila anda ingin merevisi tugas, buat release dengan penambahan angka sebelumnya.
       Contoh, kelompok anda sudah merilis v1, namun ternyata ada revisi sedikit, silakan buat rilis v2.
       Tugas yang dinilai adalah release versi terakhir.
    c. Waktu pengumpulan tugas yang dilihat adalah **waktu release version terakhir ke server Gitlab** terakhir.
       Akan ada pengurangan terhadap release version yang melebihi waktu deadline tugas.
- Alasan deadline jam 3 sore adalah pencegahan apabila terjadi error pada server Gitlab Informatika. Karena pada jam tersebut masih ada staff yang dapat menangani hal tersebut.
- (Opsional, hanya saran) Silakan buat git cadangan ke platform lain seperti *github*, *gitlab* (bukan yang gitlab informatika) atau *bitbucket*, sebagai git cadangan apabila gitlab informatika ada masalah. Sehingga, apabila gitlab informatika sedang mati sementara atau *down* masih bisa mengerjakan.
    a. Caranya, tinggal tambah remote repository menggunakan command git.
    b. Saat git push, targetkan ke remote repository cadangan.
    c. Kalau git pull, targetkan ke remote repository yang memiliki *commit* paling baru.

Pembagian Kelompok

Kelompok maksimal terdiri dari 3 orang, *tidak wajib berjumlah 3 orang*, diperbolehkan lintas kelas. Daftarkan kelompok kalian maksimal **Jumat, 7 Oktober 2022 pukul 23.59 WIB**. Pengisian kelompok ada di ~~sheet ini~~ teams.

Pembagian Tugas

Setiap anggota kelompok **diwajibkan** untuk mengerjakan bagian *server-side* (PHP) dan *client-side* (HTML, CSS, JS), dengan harapan kalian bisa mencoba mempelajari semua bagian dan cara kerjanya.

Server-side

Login : 13520xxx, 13520xxx

Register : 13520xxx

(Lanjutkan ...)

Client-side

Login : 13520xxx, 13520xxx

Register : 13520xxx

(Lanjutkan...)

Catatan Tambahan

Tugas ini akan terdiri dari dua milestone. Milestone ini akan berhubungan dengan milestone selanjutnya; artinya tugas yang kalian buat di sini **akan digunakan kembali pada milestone berikutnya**. Usahakan untuk menyelesaikan spesifikasi utama agar tidak terhambat pada milestone berikutnya.

Sangat disarankan *(tapi tidak memaksa kok)* untuk mengerjakan **Bonus 2 (Docker)**, karena akan membantu **memudahkan** kalian untuk milestone berikutnya dan mata kuliah yang akan datang.