# Tugas Project Setup BDD



**Sebelumnya apakah prettier itu?**

Prettier adalah tools pemformatan kode yang dapat digunakan Developer untuk memformat kode mereka secara otomatis menggunakan pedoman yang telah ditetapkan sebelumnya. Beberapa pedoman ini dapat disesuaikan dengan menggunakan file konfigurasi yang Lebih baik dan bagus. Misalnya, developer dapat menggunakan file konfigurasi untuk menyetel apakah kode yang diformat akan menyertakan tanda kutip tunggal atau tanda kutip ganda, atau apakah kode akan menyertakan titik koma di akhir setiap pernyataan.

Beberapa file konfigurasi Prettier disimpan sebagai file PRETTIERRC. File ini berisi daftar setelan pengguna yang Lebih teratur , yang diformat menggunakan JSON atau YAML. File konfigurasi Lebih teratur lainnya mungkin berisi daftar pengaturan berformat JavaScript atau TOML.

CATATAN: File PRETTIERRC juga dapat berisi pengaturan yang diganti, digunakan untuk menimpa pengaturan saat memformat kode yang disimpan di folder atau file tertentu.

**Membangun dan menegakkan panduan gaya**

Sejauh ini alasan terbesar untuk mengadopsi Prettier adalah untuk menghentikan semua perdebatan yang sedang berlangsung tentang gaya. Secara umum diterima bahwa memiliki panduan gaya umum sangat berharga untuk sebuah proyek dan tim, tetapi mencapainya adalah proses yang sangat menyakitkan dan tidak menguntungkan. Orang-orang menjadi sangat emosional dengan cara-cara tertentu dalam menulis kode dan tidak ada yang suka menghabiskan waktu menulis dan menerima telur kutu.

Jadi mengapa memilih "Panduan gaya yang lebih cantik" daripada panduan gaya acak lainnya? Karena Prettier adalah satu-satunya “panduan gaya” yang sepenuhnya otomatis. Bahkan jika Prettier tidak memformat semua kode 100% seperti yang Anda inginkan, itu sepadan dengan "pengorbanan" mengingat manfaat unik dari Prettier, bukan begitu?

* “Kami ingin membebaskan utas mental dan mengakhiri diskusi seputar gaya. Meskipun terkadang berbuah, diskusi ini sebagian besar sia-sia.”
* “Secara harfiah memiliki seorang insinyur melalui upaya besar untuk membersihkan semua kode kami karena kami memperdebatkan gaya ternary untuk waktu yang lama dan tidak konsisten tentang hal itu. Itu bodoh, tapi itu adalah "debat hebat" aneh yang sedang berlangsung yang membuang banyak sedikit bolak-balik. Jauh lebih mudah bagi kita semua untuk setuju sekarang: jalankan Prettier, dan ikuti gaya itu.”
* “Menjadi lelah memberi tahu orang-orang cara menata kode produk mereka.”
* “Alasan utama kami adalah untuk berhenti membuang waktu berdebat tentang gaya rambut.”
* “Memiliki pengaturan githook telah mengurangi jumlah masalah gaya dalam PR yang mengakibatkan build rusak karena aturan ESLint atau hal-hal yang harus saya pilih atau bersihkan nanti.”
* "Saya tidak ingin ada orang yang mengganggu orang lain lagi."
* “Itu mengingatkan saya pada bagaimana Steve Jobs dulu memakai pakaian yang sama setiap hari karena dia memiliki sejuta keputusan untuk dibuat dan dia tidak ingin direpotkan dengan hal-hal sepele seperti memilih pakaian. Saya pikir Lebih Cantik seperti itu. ”

**Membantu Pendatang Baru**

Prettier biasanya diperkenalkan oleh orang-orang dengan pengalaman dalam basis kode dan JavaScript saat ini, tetapi orang-orang yang mendapat manfaat secara tidak proporsional darinya adalah pendatang baru di basis kode. Orang mungkin berpikir bahwa itu hanya berguna untuk orang-orang dengan pengalaman pemrograman yang sangat terbatas, tetapi kami telah melihatnya mempercepat waktu peningkatan dari para insinyur berpengalaman yang bergabung dengan perusahaan, karena mereka mungkin menggunakan gaya pengkodean yang berbeda sebelumnya, dan pengembang yang berasal dari pemrograman yang berbeda. bahasa.

* “Motivasi saya menggunakan Prettier adalah: terlihat bahwa saya tahu cara menulis JavaScript dengan baik.”
* “Saya selalu meletakkan spasi di tempat yang salah, sekarang saya tidak perlu khawatir lagi.”
* “Ketika Anda seorang pemula, Anda membuat banyak kesalahan yang disebabkan oleh sintaks. Berkat Prettier, Anda dapat mengurangi kesalahan ini dan menghemat banyak waktu untuk fokus pada hal yang benar-benar penting.”
* “Sebagai seorang guru, saya juga akan memberi tahu siswa saya untuk menginstal Prettier untuk membantu mereka mempelajari sintaks JS dan memiliki file yang dapat dibaca.”

**Menulis kode**

Apa yang biasanya terjadi setelah orang menggunakan Prettier adalah bahwa mereka menyadari bahwa mereka sebenarnya menghabiskan banyak waktu dan energi mental untuk memformat kode mereka. Dengan integrasi editor yang lebih cantik, Anda cukup menekan kunci ajaib yang mengikat dan poof, kode diformat. Ini adalah pengalaman yang membuka mata jika ada yang lain.

* “Saya ingin menulis kode. Tidak menghabiskan siklus untuk memformat.”
* “Ini menghapus 5% yang menyebalkan dalam kehidupan kita sehari-hari - alias pemformatan”
* “Kami berada di tahun 2017 dan masih menyakitkan untuk memecah panggilan menjadi beberapa baris ketika Anda menambahkan argumen yang membuatnya melampaui batas 80 kolom :(“

**Mudah di terapkan**

Kami telah bekerja sangat keras untuk menggunakan gaya pengkodean yang paling tidak kontroversial, melewati banyak putaran untuk memperbaiki semua kasus tepi dan memoles pengalaman memulai. Ketika Anda siap untuk mendorong Prettier ke dalam basis kode Anda, Anda tidak hanya harus melakukannya secara teknis tanpa rasa sakit, tetapi basis kode yang baru diformat seharusnya tidak menimbulkan kontroversi besar dan diterima tanpa rasa sakit oleh rekan kerja Anda.

* “Ini overhead yang rendah. Kami dapat melempar Prettier pada jenis repo yang sangat berbeda tanpa banyak usaha.”
* “Sebagian besar sudah bebas bug. Jika ada masalah gaya utama selama implementasi, kami akan berhati-hati untuk membuang ini ke basis kode JS kami. Saya senang untuk mengatakan bahwa bukan itu masalahnya.”
* “Semua orang menjalankannya sebagai bagian dari skrip pra-komit mereka, beberapa dari kami juga menggunakan editor pada ekstensi penyimpanan.”
* “Cepat, melawan salah satu basis kode JS kami yang lebih besar, kami dapat menjalankan Prettier dalam waktu kurang dari 13 detik.”
* “Manfaat terbesar bagi Prettier bagi kami adalah dapat memformat seluruh basis kode sekaligus.”

**Membersihkan basis kode yang ada**

Karena memunculkan gaya pengkodean dan menegakkannya adalah tugas besar, ia sering lolos dari celah dan Anda dibiarkan bekerja pada basis kode yang tidak konsisten. Menjalankan Lebih Cantik dalam hal ini adalah kemenangan cepat, basis kode sekarang seragam dan lebih mudah dibaca tanpa menghabiskan banyak waktu.

* "Lihat kodenya :) Saya hanya perlu mengembalikan kewarasan."
* “Kami mewarisi ~2000 modul basis kode ES6, yang dikembangkan oleh 20 pengembang berbeda selama 18 bulan, dalam tim global. Merasa seperti kemenangan tanpa banyak penelitian.”

**Naik kereta hype**

Aspek teknis murni dari proyek bukanlah satu-satunya hal yang diperhatikan orang ketika memilih untuk mengadopsi Prettier. Siapa yang membangun dan menggunakannya dan seberapa cepat itu menyebar melalui komunitas memiliki dampak yang tidak sepele.

* “Hal yang luar biasa bagi saya adalah: 1) Diumumkan 2 bulan lalu. 2) Sudah diadopsi oleh, tampaknya, setiap proyek JS besar. 3) 7000 bintang, 100.000 unduhan npm/bln”
* “Dibangun oleh orang yang sama dengan React & React Native.”
* “Saya suka menjadi bagian dari hal-hal baru yang hangat.”
* "Karena tidak lama lagi orang akan memintanya."

**Kesimpulan tentang prettirrc**

Setelah menggali beberapa informasi sedikit tentang apa itu prettierrc dan beberapa hal diatas merupakan sebagian yang saya kutip, jadi menanggapi hal apakah file prettierrc itu penting atau tidak, kesimpulan saya berdasarkan pada informasim yang saya dapat.

“File prettierrc ternyata cukup penting terutama dalam penerapan pengkodean oleh para pengembang secara global, ini karena file prettierrc ini dikembangkan untuk menjadi solusi tentang menerapan gaya kode yang berbeda dari pada basis yang seringkali diperdebatkan, khususnya oleh para developers yang barangkali membuat jalannya pengembang jadi kurang efisien. Itulah mengapa saya menganggap file prettier cukup penting selain dari segi fungsi yang didapat ternyata secara tidak langsung ini juga dapat mencegah masalah internal yang menyakitkan dalam perdebatan yang melibatkan banyak developers secara global”