**TUGAS KELOMPOK**

**PHALCON**

****

**DISUSUN OLEH :**

**SILVY DHARMA FEBRYANA**

**EMY SAFITRI**

**PUTRI AULIA**

**IKHSAN ABDUL HADI NUGRAGA**

**LIBRANTARA ERLANGGA**

**D4 TEKNIK INFORMATIKA 1D**

**POLITEKNIK POS INDONESIA**

**BANDUNG**

**2015-2016**

**Mengenal Phalcon Team: Developer Inti PhalconPHP**

Phalcon memiliki dokumentasi yang baik dan jelas. Belum lagi dokumentasinya sudah mulai ditranslasikan ke dalam Bahasa lain seperti Indonesia, Jepang, Belanda , Polandia, , Perancis, Thai, dan lainnya. kini Phalcon sudah [rilis di versi 2.0.3 pada Juni 2016](http://www.codepolitan.com/rilisan-terbaru-phalcon-2-0-3/) ini. PhalconPHP sendiri merupakan *web framework* yang bersifat *open source*. . Banyak *developer* yang ikut berkontribusi dalam pengembangan Phalcon untuk menjadi *web framework* tercepat dan terhandal. Namun semua v itu tidak lepas dari *developer* inti yang berada dibalik layar Phalcon.

Siapa sajakah para *developer* inti dibalik Phalcon?

**Andres Gutierrez**

**Eduar Carvajal**

**Nikolaos Dimopoulos**

**Nikolay Kirsh**

Phalcon adalah sebuah framework PHP yang bersifat open-source yang dibangun dengan ekstensi dari C. Tapi tenang kamu tidak perlu menguasai bahasa C untuk menggunakan framework ini karena fungsionalitas dari C itu sendiri telah disajikan dalam class-class yang mudah digunakan dalam bahasa pemrograman PHP. Team Phalcon mengklaim bahwa framework ini memiliki performa yang tinggi dan sangat cepat, karena mengkonsumsi resource yang sangat sedikit. Selain itu, framework ini dianggap memiliki fitur yang kaya dan mudah digunakan. Framework ini mengadopsi prinsip OO

[](http://4.bp.blogspot.com/-azBZi7zInjs/VKPBis7WSPI/AAAAAAAAAIU/QS0IA6oIXW0/s1600/fast.png)Performa berbagai PHP Framework

(Object Oriented). Kebanyakan framewok yang berbasis OO membutuhkan keterlibatan dan *inheritance* dari banyak file dan ratusan baris code program untuk dieksekusi, yang membuat kebanyakan framework ini membawa beban eksekusi yang berlebih sehingga aplikasi-aplikasi komplex yang dibangun menjadi lebih lambat. Sebagai gambaran, pada framework umumnya, saat aplikasi/web telah dijalankan, untuk memenuhi 1x request dibutuhkan hingga ratusan file (bisa sampai 300 file!) yang di-*include*. Meski pada kenyataannya hanya 25% dari jumlah tersebut yang merupakan kode yang kita buat!