The Cathedral and The Bazaar

Setelah membaca artikel mengenai The Cathedral and The Bazaar yang membahas mengenai dua model dasar yang sangat berlawanan dalam hal software-debugging. The Cathedral merupakan suatu model di mana disediakannya source code dari sebuah software yang di-releases namun, pengembangannya masih dibatasi dan hanya bisa diakses oleh para software-developers. Sedangkan untuk model Bazaar di mana coding sebuah software dikembangkan melalui jaringan Internet dan bisa dilihat dan diakses secara public, dan model Bazaar ini sebenarnya diciptakan oleh Linus Torvalds, pemimpin dari Linux kernel project. Linux selalu menggunakan model Bazaar ini dalam mengembangan berbagai software melalui menyebarkan source code milik setiap software secara luas agar orang-orang bahkan orang awam pun bisa mencoba melakukan percobaan pada software tersebut seperti system keamanannya, menemukan segala bentuk bugs dan memperbaiki, serta banyak percobaan lain dapat dilakukan oleh banyak orang terhadap software tersebut. Hal ini tentunya memiliki keuntungan besar yaitu software dapat dikembangkan dengan jauh lebih cepat dan sempurna daripada mengembangkannya dengan menggunakan model Cathedral yang menutup akses pengembangan software hanya untuk orang-orang tertentu saja, yang meskipun expert tapi sering kali belum tentu dapat menemukan dan memperbaiki masalah seperti bugs dalam software dengan cepat.

Saat membaca artikel ini, saya menemukan 19 poin penting yang ditulis oleh Raymond dalam hal bagaimana menciptkan open source software yang bagus, di mana semua poin ini diperoleh setelah melakukan pengamatan dan penelitian terhadap berbagai cara pengembangan software open source. Ada beberapa poin yang menurut saya sangat menarik, seperti :

1. Good programmers know what to write. Great ones know what to rewrite and reuse.

Poin ini menarik sebab memang benar bahwa dalam mengembangkan software bukan hanya diperlukan kemampuan untuk menulis coding yang baik di mana kemampuan ini memang penting bagi semua programmer. Akan tetapi bila punya kemampuan untuk menulis ulang bahkan menggunakannya kembali, kemampuan itu akan mampu membuat pengembangan software open source makin cepat dan menjadi lebih sempurna.

1. [Release early. Release often.](https://en.wikipedia.org/wiki/Release_early,_release_often) And listen to your customers

Setelah selesai mengembangkannya pastilah software harus didistribusikan. Tapi lebih baik lebih cepat didistribusikan agar kita bisa mengetahui apa saja yang mungkin kurang dan masih bisa dikembangkan lebih jauh lagi dari software tersebut dari percobaan-percobaan yang dilakukan oleh para customer. Kemudian setelah itu, software pun bisa di-maintenance dan dikembangakan sesuai pendapat atau perbaikan oleh para customer, untuk kemudian didistribusikan kembali. Hali ini pun akan terjadi berulang-ulang seperti maintenance dan pendistribusian ulang.

1. Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.

Dalam mengembangkan software, sangatlah penting untuk membiarkan para beta tester untuk melakukan percobaan agar bisa melihat apa ada kekurangan seperti bugs atau area apa yang masih bisa dikembangkan lebih jauh dari sebuah software seperti software games atau open source software lain seperti Open Office. Orang-orang yang berpengalaman dan paham dengan dunia pengembangan software seperti beta tester dan co-developer base akan mampu memahami permasalahan dalam software dan memperbaikinya dengan cepat.

1. To solve an interesting problem, start by finding a problem that is interesting to you.

Poin ini juga sangat menarik sebab setiap softwaree tidaklah selalu sempurna, oleh karena itu kita pun bisa mencoba untuk mencari suatu problem dari sebuah software dan mencoba untuk memperbaikinya. Dengan hali itu, kemampuan kita pun akan berkembang cepat dan bisa memberikan kontribusi besar dalam membantu mengembangkan sofware.

Dengan ini saya dapat memberikan kesimpulan bahwa model Bazaar merupakan model yang sangat efektif dan menguntungkan banyak pihak seperti pengembangnya dan customers. Kentungannya yaitu pertama, pengembangan software cepat dan makin sempurna atau sesuai dengan keinginan banyak orang, Kedua mampu mengembangkan kemampuan programming.