Belajar Git: Dasar-dasar Pengelolaan Repository

Repository pada dasarnya merupakan gudang tempat menyimpan semua berkas yang kita miliki.

**Membuat Repository Baru**

1. Login ke github, klik pada tombol "New Repository" untuk membuat repository baru, dan isi informasi yang diminta. Setelah repository dibuat, yang perlu diperhatikan adalah alamat yang diberikan untuk repository tersebut. Misalnya untuk saya (yang username di github adalahrahmadd) dengan membuat sebuah repository yang benama repoku maka alamat yang akan diberikan adalah: rahmadd/repoku
2. Buka terminal, buat direktori tempat kita menyimpan repository kita, dan masuk ke dalam direktori tersebut. Misalkan untuk contoh repository di atas:

|  |  |
| --- | --- |
| 1  2 | mkdir repoku  cd repoku |

1. Buat sebuah berkas README.md yang akan berisi keterangan singkat tentang perangkat lunak ini:

|  |  |
| --- | --- |
| 1 | touch README.md |

1. Buka berkas ini dan ketikan keterengan singkat tentang perangkat lunak ini.
2. Inisialisasi git pada direktori ini:

|  |  |
| --- | --- |
| 1 | git init |

1. Hubungkan direktori ini dengan repository yang baru kita buat di github. Misalkan untuk contoh repository di atas:

|  |  |
| --- | --- |
| 1  2 | git remote add origin git@github.com<script cf-hash="f9e31" type="text/javascript">  /\* <![CDATA[ \*/!function(){try{var t="currentScript"in document?document.currentScript:function(){for(var t=document.getElementsByTagName("script"),e=t.length;e--;)if(t[e].getAttribute("cf-hash"))return t[e]}();if(t&&t.previousSibling){var e,r,n,i,c=t.previousSibling,a=c.getAttribute("data-cfemail");if(a){for(e="",r=parseInt(a.substr(0,2),16),n=2;a.length-n;n+=2)i=parseInt(a.substr(n,2),16)^r,e+=String.fromCharCode(i);e=document.createTextNode(e),c.parentNode.replaceChild(e,c)}}}catch(u){}}();/\* ]]> \*/</script>:rahmadd/repoku.git |

1. Uraian perintah-perintah ini:
   * remote add merupakan perintah git untuk menambahkan (add) satu kopian dari suatu repository di github (dan karena letaknya di server nun jauh berarti bersifat tidak lokal atau remote);
   * origin merupakan nama yang kita berikan untuk kopian dari repository ini;
   * git@github.com: merupakan perintah untuk membuka koneksi git ke github.com;
   * rahmadd/repoku.git merupakan alamat berkas dari repository di github yang telah kita buat sebelumnya (lihat langkah 1). Alamat ini sama dengan alamat repository kita tetapi ditambah akhiran .git.
2. Tambahkan semua berkas dan subdirektori ke repository kita dengan perintah:

|  |  |
| --- | --- |
| 1 | git add . |

1. Commit, atau perintahkah git untuk mencatat segala perubahan yang telah kita buat dengan menjalankan:

|  |  |
| --- | --- |
| 1 | git commit -m "Commit pertama" |

1. Uraian perintah-perintah ini:
   * commit merupakan perintah git untuk mencatat segala perubahan yang terjadi;
   * -m "Commit pertama" merupakan pesan/komentar yang hendak kita kaitkan dengan semua perubahan ini.
2. Kirim semua yang telah kita commit ke github:

|  |  |
| --- | --- |
| 1 | git push origin master |

Uraian perintah ini:

* + push merupakan perintah git untuk mengirim (tolak=push) perubahan yang telah kita tandai sebelumnya dengan perintah commit (lihat langkah 7) ke github;
  + origin merupakan nama dari kopian repository yang ada pada mesin kita (lihat langkah 5);
  + master merupakan nama dari cabang utama kode kita di github. Semua cabang utama di github akan diberi nama master.

**Mengirim Perubahan ke Repository**

1. Jika anda telah selesai mengubah kode anda dan siap untuk dikirim ke server maka lakukan perintah-perintah berikut:

|  |  |
| --- | --- |
| 1  2  3 | git add .  git commit -m "Komentar tetang perubahan yang dilakukan"  git push origin master |

1. Perhatikan jika perintah ini meminta password maka ini password untuk akun adan di Github.
2. Perintah pertama (add) akan memeriksa semua kode pada directory saat ini (.) anda dan menandai yang mana saja yang telah mengalami perubahan;
3. Perintah kedua (commit) akan memcatat segala perubahan yang terjadi pade semua berkas dan siap untuk dikirim ke server. Opsi -m "Komentar tetang perubahan yang dilakukan" merupakan komentar yang akan dikaitkan dengan segala perubahan ini;
4. Perintah ketiga (push) adalah proses pengiriman perubahan-perubahan dari kopian repository kita (yang sebelumnya kita beri nama origin) ke cabang utama di server (yang secara default diberi nama master).

**Menghapus Berkas dari Repository**

1. Jika anda hendak menghapus suatu berkas dari repository anda maka lakukan perintah-perintah berikut:

|  |  |
| --- | --- |
| 1  2  3 | git rm nama-berkas  git commit -m "Komentar tetang perubahan yang dilakukan"  git push origin master |

1. Perintah pertama (rm) akan menghapus berkas dengan nama nama-berkas;
2. Perintah kedua (commit) akan memcatat perubahan yang terjadi;
3. Perintah ketiga (push) adalah proses pengiriman perubahan ke server.

**Mengganti Nama Berkas/Directory yang Sudah Ada di Repository**

1. Jika anda hendak ganti nama suatu berkas atau directory yang sudah ada di repository anda maka lakukan perintah-perintah berikut:

|  |  |
| --- | --- |
| 1  2  3 | git mv nama-lama nama-baru  git commit -m "Komentar tetang perubahan yang dilakukan"  git push origin master |

1. Perintah pertama (mv) akan mengganti nama berkas atau directory lama anda ke nama baru;
2. Perintah kedua (commit) akan memcatat perubahan yang terjadi;
3. Perintah ketiga (push) adalah proses pengiriman perubahan ke server.

**Membuat Salinan dari Repository yang Sudah Ada**

1. Misalkan kita ingin mengkopi repository repoku yang telah kita buat di atas ke directory lain misalnya benarma repolain;
2. Buka terminal, buat direktori tempat kita akan menyimpan repository kita, dan masuk ke dalam direktori tersebut:

|  |  |
| --- | --- |
| 1  2 | mkdir repolain  cd repolain |

1. Perintahkan git untuk mengkopi repository repoku ke directory ini dengan menjalankan:

|  |  |
| --- | --- |
| 1  2 | git clone git@github.com<script cf-hash="f9e31" type="text/javascript">  /\* <![CDATA[ \*/!function(){try{var t="currentScript"in document?document.currentScript:function(){for(var t=document.getElementsByTagName("script"),e=t.length;e--;)if(t[e].getAttribute("cf-hash"))return t[e]}();if(t&&t.previousSibling){var e,r,n,i,c=t.previousSibling,a=c.getAttribute("data-cfemail");if(a){for(e="",r=parseInt(a.substr(0,2),16),n=2;a.length-n;n+=2)i=parseInt(a.substr(n,2),16)^r,e+=String.fromCharCode(i);e=document.createTextNode(e),c.parentNode.replaceChild(e,c)}}}catch(u){}}();/\* ]]> \*/</script>:rahmadd/repoku.git . |

1. Perhatikan jika perintah ini meminta password, ini merupakan password untuk mengakses SSH key yang anda buat sebelumnya (ingat kembali proses instalasi di tulisan [ini](http://rahmaddawood.blogspot.com/2012/02/belajar-git-instalasi-dan-konfigurasi.html)). Uraian perintah ini:
   * clone merupakan perintah git untuk membuat kopian dari repository;
   * git@github.com: merupakan perintah untuk membuka koneksi git ke github.com;
   * rahmadd/repoku.git merupakan alamat dari repository yang hendak kita kopi dari github (ini sama dengan alamat repository kita tetapi ditambah akhiran .git);
   * . titik pada akhir perintah ini menandai untuk melakukan proses copy ke directory ini. Seandainya kita hendak membuat kopian di drectory xyz maka perintahnya menjadi:

|  |  |
| --- | --- |
| 1  2 | git clone git@github.com<script cf-hash="f9e31" type="text/javascript">  /\* <![CDATA[ \*/!function(){try{var t="currentScript"in document?document.currentScript:function(){for(var t=document.getElementsByTagName("script"),e=t.length;e--;)if(t[e].getAttribute("cf-hash"))return t[e]}();if(t&&t.previousSibling){var e,r,n,i,c=t.previousSibling,a=c.getAttribute("data-cfemail");if(a){for(e="",r=parseInt(a.substr(0,2),16),n=2;a.length-n;n+=2)i=parseInt(a.substr(n,2),16)^r,e+=String.fromCharCode(i);e=document.createTextNode(e),c.parentNode.replaceChild(e,c)}}}catch(u){}}();/\* ]]> \*/</script>:rahmadd/repoku.git xyz |

* + Perhatikan bahwa kopian dari repository ini akan diberi nama origin.

**Menarik Perubahan dari Repository**

1. Agar anda mendapatkan kode terbaru yang telah dibuat oleh anggota-anggota tim anda maka lakukan perintah-perintah berikut:

|  |  |
| --- | --- |
| 1 | git pull |

1. Perintah alternatif adalah:

|  |  |
| --- | --- |
| 1  2 | git fetch  git merge origin |

1. Perintah pertama (fetch) bertugas mengambil semua perubahan yang terjadi dari server semenjak perintah fetch terakhir;
2. Perintah kedua (merge) menggabungkan perubahan-perubahan yang terjadi ke kopian repository kita (yang secara default akan diberi nama origin saat kita kopikan repository tersebut dengan perintah clone).

**Melihat Perbedaan dengan Repository**

Untuk melihat perbedaan antara bekas yang anda miliki dengan yang saat ini tersimpan di server jalankan perintah berikut:

|  |  |
| --- | --- |
| 1 | git diff |