**MATEMATIKA DISKRIT**

Tugas Pertemuan 3

Jenis Tugas : **Rangkuman**

Nama Kelompok : **HAPPY BEAR**

1. Yusuf Fataa Muhana Lathif Al Huda

2. Belinda Sukma

3. Ihsan Setiyaadi

4. Ambarsari

**I. HIMPUNAN**

Jenis- Jenis Himpunan: 1. Himpunan Semesta

2. Himpunan Kosong

3. Himpunan Bagian (Subset)

4. Himpunan Sama

5. Himpunan Ekivalen

6. Himpunan Saling Lepas

7. Himpunan Kuasa

Dalam kehidupan sehari-hari, kita pasti akan menemukan atau setidaknya mengenal suku Jawa, suku Madura, suku Batak, dll. Semua nama-nama suku itu merupakan kelompok.

Mengutip modul Matematika Kemdikbud karya Abdur Rahman As’ari, dkk, istilah kelompok, kumpulan, golongan, maupun gerombolan dalam matematika dikenal dengan istilah himpunan. Teori Himpunan ditemukan oleh seorang ahli matematika asal Jerman, bernama Georg Cantor (1845-1918).

Suatu himpunan dapat dinyatakan dengan menyebutkan semua anggotanya, dengan dituliskan dalam kurung kurawal "{}". Apabila, banyak anggotanya sangat banyak, maka cara mendaftarkannya biasanya dimodifikasi, dengan diberi tanda tiga titik ("...") dengan pengertian "dan seterusnya mengikuti pola".

Himpunan dapat dinyatakan dengan menyebutkan sifat yang dimiliki anggotanya.  
  
Menuliskan syarat keanggotaan himpunan tersebut. Notasi ini biasanya berbentuk umum {x | P(x)}, dimana x mewakili anggota dari himpunan, dan P(x) menyatakan syarat yang harus dipenuhi oleh x agar bisa menjadi anggota dari himpunan tersebut. Simbol x bisa diganti oleh variabel yang lain, seperti y, z, dan lain-lain. Misalnya, A = {1, 2, 3, 4, 5} bisa dinyatakan dengan notasi pembentuk himpunan A = {x | x < 6, dan x ∈ asli}.

**1. Himpunan Semesta**

Himpunan yang melingkupi semua anggota dalam daerah pembicaraan.

Istilah lain dari Himpunan Semesta adalah Himpunan Universal (disimbolkan dengan U).

Himpunan semesta dinotasikan dengan (S). Untuk mengetahui tentang himpunan semesta, kita perlu mengetahui himpunan dan anggota-anggota di dalamnya. Misalnya, ada tiga himpunan beserta anggotanya, yakni A = {anjing, kelinci, kucing}, B = {hiu, paus, lumba-lumba}, C = {elang, merpati, burung beo}.  
  
Jika kita amati, himpunan A merupakan nama-nama hewan yang biasanya dipelihara, sedangkan himpunan B adalah nama-nama hewan yang hidupnya di laut, dan himpunan C adalah nama-nama hewan yang terbang. Bisa dipastikan himpunan semesta dari ketiga unsur himpunan A, B, dan C adalah nama hewan. Jadi, himpunan semestanya dapat ditulis dengan S = {nama hewan}.  
  
Contoh Soal:  
Tentukan himpunan semesta yang mungkin dari himpunan-himpunan berikut.  
A = {pesawat terbang, kapal, motor, mobil, kereta }  
B = {pisang, salak, durian, mangga}  
C = {16, 25, 36, 49} 4.  
D = {−2, −1, 0, 1, 2, 3,4, 5, 6}  
  
Jawaban:  
Himpunan semesta (S) dari anggota himpunan:  
A = {himpunan alat transportasi}  
B = {himpunan buah}  
C = {himpunan bilangan kuadrat 10 dan 50}  
D = {himpunan bilangan bulat antara −3 dan 7}  
  
Contoh lain:

A = {1, 3, 5}

Maka Himpunan Semesta dapat diambil U = {1, 2, 3, 4, 5}

Atau U = semua bilangan asli atau U = {-1, 0, 1, 2, 3, 4, 5}

**2. Himpunan Kosong**

himpunan kosong adalah himpunan yang tidak memiliki anggota dan penulisan lambangnya adalah {}.  
Berdasarkan buku Dasar-dasar Matematika dan Sains yang ditulis oleh Ali Nugraha dan Dina Dwiyana, himpunan kosong adalah himpunan yang anggotanya benar-benar tidak ada.  
-himpunan {{ }} dapat juga ditulis sebagai { Ø }

-himpunan {{ }, {{ }}} dapat juga ditulis sebagai { Ø, { Ø }}

-{ Ø } bukan himpunan kosong karena ia memuat satu elemen yaitu himpunan kosong.

**3. Himpunan Bagian**

adalah himpunan yang seluruh anggota berada di himpunan lain. Unsur-unsur himpunan bisa berupa apa saja seperti sekelompok bilangan real, variabel, konstanta, bilangan bulat, dll. Ini juga terdiri dari himpunan nol.  
Simbol himpunan bagian yaitu ⊂ artinya "himpunan bagian dari", sedangkan ⊄ artinya "bukan himpunan dari".

Contoh 1:  
A = {13, 15, 17}  
B = {13, 14, 15, 16, 17}  
Disini himpunan A merupakan bagian dari himpunan B maka A ⊂ B karena anggota A juga merupakan anggota B.  
  
  
Contoh 2  
A = {1,2,3}  
B = {1,2,3,4,6}  
C = {8,9,10}  
Dapat diketahui himpunan A merupakan bagian dari himpunan B atau kita tuliskan dengan simbol A ⊂ B. Hal ini juga artinya himpunan B adalah superset dari himpunan A atau disimbolkan dengan B ⊃ A.  
Nah, anggota himpunan C tidak ada dalam himpunan A atau B sehingga himpunan C bukan bagian dari himpunan A (C ⊄ A) juga bukan himpunan B (C ⊄ B).

**4. Himpunan Sama**

Dua buah himpunan dikatakan sama apabila kedua himpunan tersebut memiliki anggota yang sama walaupun urutannya dapat berbeda.

A = B jika dan hanya jika setiap elemen A merupakan elemen B dan sebaliknya setiap elemen B merupakan elemen A.

A = B jika A adalah himpunan bagian dari B dan B adalah himpunan bagian dari A.

Jika tidak demikian, maka A ≠ B.

Notasi : A = B ↔ A ⃀ B dan B ⃀ A

(i) Jika A = { 0, 1 } dan B = { x | x (x – 1) = 0 }, maka A=B

(ii) Jika A = { 3, 5, 8, 5 } dan B = {5, 3, 8 }, maka A = B

(iii) Jika A = { 3, 5, 8, 5 } dan B = {3, 8}, maka A ≠ B

**5. Himpunan Ekivalen**

Dalam Matematika, himpunan dapat disebut ekuivalen **jika jumlah anggota kedua himpunan sama namun bendanya ada yang tidak sama**. Dengan kata lain, dua himpunan A dan B bisa dikatakan sebagai ekuivalen jika anggota himpunan A memiliki jumlah yang sama dengan anggota himpunan B. Notasi dari ekuivalen, yakni n(A) = n(B).

Himpunan A dikatakan ekivalen dengan himpunan B

jika dan hanya jika kardinalitas dari kedua

himpunan tersebut sama.

Notasi : A ~ B ↔ |A| = |B|

Contoh

Misalkan A = { 1, 3, 5, 7 } dan B ={ a, b, c, d },

maka A ~ B sebab |A| = |B| = 4

**6. Himpunan Saling Lepas**

Dalam [matematika](https://id.wikipedia.org/wiki/Matematika), dua [himpunan](https://id.wikipedia.org/wiki/Himpunan) dikatakan saling lepas (atau saling pisah atau saling asing) jika keduanya tidak memiliki [anggota](https://id.wikipedia.org/wiki/Elemen_(matematika)) persekutuan. Sebuah keluarga himpunan adalah saling terlepas pasang demi pasang jika setiap dua himpunan berbeda dalam keluarga tersebut adalah terlepas.

Perhatikan himpunan-himpunan berikut:

A = {1, 2, 5}  
B = {3, 4, 6, 8}

Adakah anggota himpunan A yang ada di dalam himpunan B? Terlihat bahwa tidak ada satupun anggota A yang terdapat pada himpunan B, begitu juga sebaliknya. Jadi, himpunan A dan B disebut himpunan saling lepas atau saling asing, ditulis A ⫗ B (dibaca “A saling lepas dengan B”).

**7. Himpunan Kuasa**

Himpunan kuasa (power set) dari himpunan A adalah suatu himpunan yang elemennya merupakan semua himpunan bagian dari A, termasuk himpunan kosong dan himpunan A sendiri.

Notasi : ℘(A) atau 2A

Jika |A| = m, maka |℘(A)| = 2m.

Contoh:

Jika A = {1,2},

maka ℘ (A) = { Ø , {1}, {2}, {1,2}}

Himpunan kuasa dari himpunan kosong adalah ℘ (Ø) = { Ø }, dan

himpunan kuasa dari himpunan { Ø } adalah ℘ ({Ø }) = { Ø, { Ø }}.

**II. GIT**

Git adalah sebuah version control system terbuka yang dikembangkan oleh Linus Torvalds pada tahun 2005. Linus Torvalds juga merupakan pengembang sistem kernel dari sistem operasi Linux.

Perbedaan GIT & GITHUB :

| GIT | GitHub |
| --- | --- |
| Dikelola oleh The Linux Foundation | Diakuisisi oleh Microsoft |
| Diakses secara offline | Diakses secara online |
| Tidak memakai fitur user management | Menggunakan fitur user management |
| Install software di penyimpanan lokal | Di-host melalui layanan cloud |
| Fokus pada version control dan code sharing | Fokus pada source code hosting yang terpusat |

**Cara membuat Repository baru di GIT**

**buat repositori baru**

buat lah direktori baru, buka dan jalankan   
git init  
untuk membuat repositori git baru.

**periksa repositori**

buat lah salinan kerja dari repositori lokal dengan menjalankan perintah  
git clone /jalur/ke/repositori  
saat menggunakan server jarak-jauh, perintahnya menjadi  
git clone namapengguna@host:/jalur/ke/repositori

**alur-kerja**

repositori lokal kamu terdiri dari tiga bagian pokok yang disebut "trees" dikelola oleh git. yang pertama adalah Direktori Kerja yang menyimpan berkas aktual. kedua adalah Indeks yang berperan sebagai pengolah data dan terakhir HEAD yang mengarah pada komit terakhir.



**tambah & komit**

kamu bisa melakukan perubahan (penambahan ke **Indeks**) menggunakan  
git add <namaberkas>  
git add \*  
Ini merupakan langkah awal alur-kerja dasar git. Untuk komit sepenuhnya gunakan  
git commit -m "Pesan komit"  
Sekarang berkas telah berkomit di **HEAD**, tapi belum di repositori jarak-jauh.

**mengirim perubahan**

Saat ini perubahan telah tersimpan di **HEAD** salinan kerja lokal kamu. Untuk mengirimkannya ke repositori jarak-jauh, lakukan   
git push origin master  
Ubah *master* sesuai cabang yang kamu inginkan.   
  
Jika repositori yang ada belum dikloning dan ingin dihubungkan ke server jarak-jauh, kamu perlu menambahkan  
git remote add origin <server>  
Sekarang kamu bisa mengirimkan perubahan ke server jarak-jauh yang dituju

**percabangan**

percabangan atau *branching* digunakan untuk mengembangkan fitur-fitur secara terisolasi. Cabang utama atau *master* merupakan cabang bawaan ketika kamu membuat repositori. Gunakan cabang lain untuk pengembangan, setelah selesai, gabungkan kembali ke cabang utama.



buat cabang baru dengan nama "fitur\_x" dan beralih kedalamnya menggunakan  
git checkout -b fitur\_x  
beralih kembali ke *master*  
git checkout master  
dan hapus cabang yang tadi dibuat  
git branch -d fitur\_x  
suatu cabang *tidak terbuka untuk yang lainnya* kecuali jika kamu mengirimkannya ke repositori jarak-jauh.  
git push origin <cabang>

**perbaru & gabung**

untuk memperbarui repositori lokal ke komit terkini, lakukan   
git pull  
dari direktori kerja kamu untuk *mengambil* dan *menggabungkan* perubahan jarak-jauh.  
untuk menggabungkan cabang lain ke cabang aktif (misal *master*), gunakan  
git merge <cabang>  
pada kasus diatas, git mencoba menggabungkan perubahan secara otomatis. Sayangnya hal ini tak selalu berjalan mulus dan bisa menyebabkan *konflik*. Kamu lah yang bertanggung jawab menggabungkan *konflik* tersebut secara manual dengan menyunting berkas yang ditunjukkan git. Setelah itu, kamu perlu memarkahinya dengan  
git add <namaberkas>  
sebelum penggabungan berlaku, kamu bisa melakukan pratinjau menggunakan  
git diff <cabang\_asal> <cabang\_tujuan>

**Pull Requests**

[Pull Requests](https://help.github.com/articles/using-pull-requests) adalah cara yang mengagumkan untuk berkontribusi pada repositori secara mandiri dengan fork itu. Pada akhirnya, jika kita mau, kita dapat mengirim permintaan tarik ke pemilik repositori untuk menggabungkan perubahan kode kita. Pull request itu sendiri dapat memicu diskusi untuk kualitas kode, fitur atau bahkan strategi umum.

**Memulai Pull Request**

Ada dua [model pull request](https://help.github.com/articles/using-pull-requests) di Github:

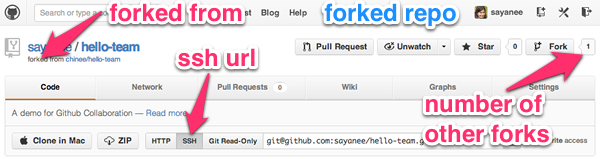
1. **Fork & Pull Model** - Digunakan di repositori publik yang tidak memiliki akses push
2. **Share Repository Model** - Digunakan dalam repositori pribadi yang kita miliki akses push. Fork tidak diperlukan adalah case ini.

Di sini kita melihat alur kerja antara dua pengguna (repo-owner dan forked-repo-owner) untuk model Fork and Pull:

1. Identifikasi Github Repository yang ingin Anda kontribusikan, dan klik tombol "Fork" untuk membuat tiruan dari repositori di akun Github Anda sendiri:



1. Ini akan membuat salinan persis dari repositori di akun Anda sendiri



1. [Pilih URL SSH](https://help.github.com/articles/why-is-git-always-asking-for-my-password) sehingga akan meminta kata kunci SSH Anda, bukan nama pengguna dan kata sandi setiap kali Anda git push atau git pull. Selanjutnya, kita akan mengkloning repositori ini ke komputer lokal:

|  |  |
| --- | --- |
| 1  2 | $ git clone [ssh-url] [folder-name]  $ cd [folder-name] |

1. Secara umum, kita akan membuat cabang git baru untuk setiap fitur baru. Ini adalah praktik yang baik karena di masa depan jika kita memperbarui cabang lebih lanjut setelah beberapa diskusi, [pull request akan diperbarui secara otomatis](http://stackoverflow.com/questions/9790448/how-to-update-a-pull-request). Mari buat cabang baru untuk membuat perubahan yang sangat sederhana untuk mengubah file: readme.md

|  |  |
| --- | --- |
| 1 | $ git checkout -b [new-feature] |

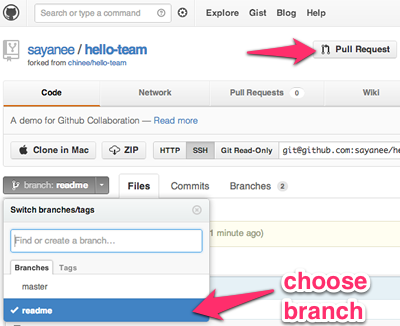
1. Setelah membuat penambahan yang relevan untuk membangun fitur-fitur baru, kita hanya akan melakukan perubahan baru dan checkout ke git master branch:

|  |  |
| --- | --- |
| 1  2  3 | $ git add .  $ git commit -m "information added in readme"  $ git checkout master |

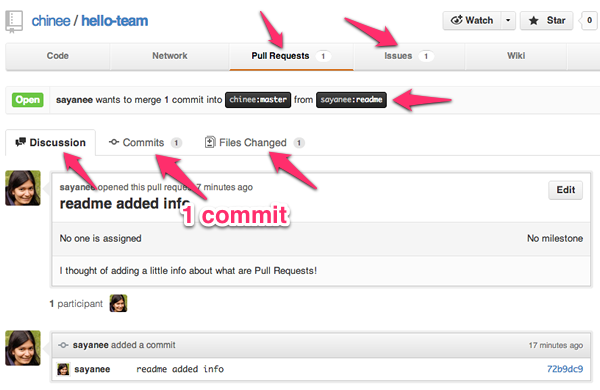
1. Pada titik ini, kita akan mendorong cabang ke repositori jarak jauh. Untuk ini kita akan memeriksa nama cabang dengan fitur baru serta alias git remote repository. Lalu kita akan mendorong perubahan menggunakan git push [git-remote-alias] [branch-name]:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $ git branch  \* master  readme  $ git remote -v  origin  git@github.com:[forked-repo-owner-username]/[repo-name].git (fetch)  origin  git@github.com:[forked-repo-owner-username]/[repo-name].git (push)  $ git push origin readme |

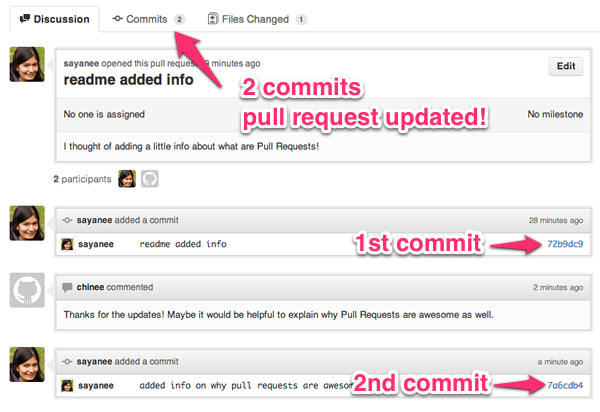
1. Di halaman Github repositori bercabang, kita akan beralih ke cabang dengan fitur baru dan kemudian tekan tombol "Pull Request".



1. Setelah mengirimkan pull request, itu akan langsung membawa kita ke halaman pull request repositori asli. Kita akan melihat pull request, baik sebagai masalah baru maupun pull request baru.



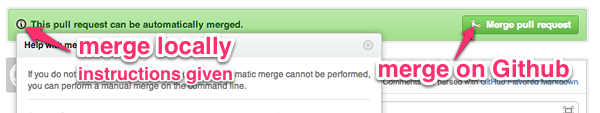
1. Setelah diskusi, mungkin pemilik repositori bercabang mungkin ingin menambahkan perubahan pada fitur baru. Dalam hal ini, kita akan melakukan pembayaran ke cabang yang sama di komputer lokal, menjalankannya, dan mendorongnya kembali ke Github. Ketika kita mengunjungi halaman pull request dari repositori asli, itu akan secara otomatis diperbarui!



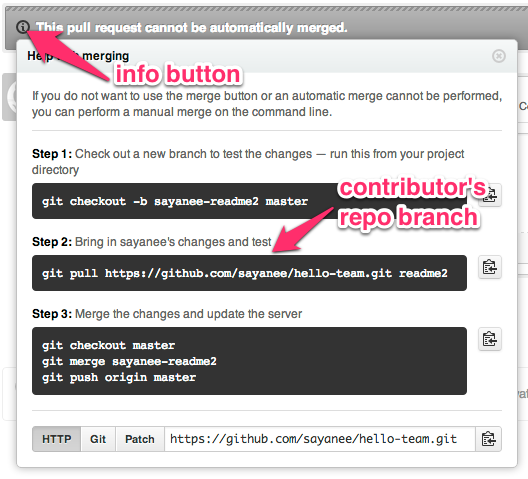
**Penggabungan Pull Request**

Jika Anda adalah pemilik repositori asli, ada [dua cara menggabungkan](https://help.github.com/articles/merging-a-pull-request) pull request masuk:

1. **Merging directly on Github:** Jika kita menggabungkan langsung di Github, maka pastikan bahwa tidak ada konflik dan siap untuk digabungkan ke dalam cabang master. Pemilik repositori asli cukup mengeklik tombol hijau "Merge Pull Request" untuk melakukannya:



1. **Merging in our local machines:** Di lain waktu, mungkin ada konflik gabungan, dan saat mengklik tombol "info", Github akan memiliki instruksi yang jelas tentang bagaimana kita dapat menggabungkan cabang di mesin lokal dengan menarik perubahan dari cabang kontributor:



Ada berbagai model percabangan yang digunakan untuk versi dalam tim pengembangan software. Berikut adalah dua model alur kerja git yang populer:

(1) [Alur kerja Github](http://scottchacon.com/2011/08/31/github-flow.html) yang memiliki model percabangan sederhana dan menggunakan pull requests dan

(2) [Gitflow](http://nvie.com/posts/a-successful-git-branching-model/) yang memiliki percabangan yang lebih luas. Model yang akhirnya dipilih pasti akan bervariasi tergantung pada tim, proyek dan situasinya.