

Pertemuan 3

Catatan :

1. PR1 >> <http://jsbin.com/baqihulozi/edit?html,js,output>
2. PR2 >> <http://jsbin.com/bulazuvipa/edit?html,js,output>
3. Sama beberapa kode terakhir harus didiskusikan ulang di momen2 santai kanal #ubuntu-indonesia

Assalamu alaikum wr.wb

waalaikumsalam warahmatullahi wabarakatuh

sebagai perwakilan dari penghuni kanal mengucapkan terima kasih atas kehadiran kawan2 semua, dalam kegiatan yang sudah kita rencanakan >> yaitu belajar bersama javascript #belajarjs

adapun pertemuan kita kali ini adalah yang ke-3 yang materinya membahas tentang **Fungsi** dan sub materi yang akan kita bahas adalah :

- 1) Deklarasi Fungsi
- 2) Fungsi Dengan Parsing Argumen
- 3) Fungsi Dengan Parsing Obyek
- 4) Fungsi Ekspresi
- 5) Fungsi Rekursif
- 6) Scope Fungsi
- 7) Stak Fungsi
- 8) Fungsi Bersarang
- 9) Konflik Nama Pada Fungsi
- 10) Closures
- 11) Obyek Sebagai Argumen
- 12) Default Parameter Dan Rest Parameter
- 13) Menyingkat Fungsi
- 14) Lexical This

Baiklah untuk mempersingkat waktu kita lanjut ke sub materi berikutnya sebelumnya ada yang mau mengasih masukan tentang definisi dari sebuah fungsi dalam bahasa pemrograman ga?

definisi fungsi apaan?

fungsi = kegunaan :-)

bisa jadi

1) Deklarasi Fungsi

Karna ga ada yang mau ngasih ide, ane coba kasih masukan >> fungsi adalah fasilitas bahasa pemrograman untuk membuat suatu perintah yang menjalankan tugas2 tertentu, dan bisa dilakukan pemanggilan berulang2.. #belajarjs

Pada javascript fungsi dapat disimpan dalam variabel, dikembalikan oleh fungsi lain, dan dikirimkan sebagai argumen untuk fungsi lainnya. Implementasi fungsi yang sangat fleksibel seperti ini membuka banyak kesempatan kepada pengembang untuk menuliskan kode yang bukan hanya berjalan dengan baik dan indah #belajarjs

deklarasi fungsi .. >> <http://pastebin.com/hk0t5isJ> <Kode1.js> #belajarjs

baiklah, untuk mempersingkat waktu kita coba latihan kode pertama #belajarjs
berikut cara membuat fungsi tanpa argumen >> <http://jsbin.com/rjjikacuga/edit?html,js,output>
<Kode2.html> #belajarjs

pada contoh diatas kita membuat sebuah fungsi buatGaris() .. sedangkan pemanggilan fungsinya ada di bagian elemen html button .. dengan memanfaatkan event onclick #belajarjs

baiklah kita melanjutkan pembahasan ke sub materi berikutnya yaitu fungsi dengan parsing argumen

2) Deklarasi Fungsi Dengan Argumen

<https://gist.github.com/anonymous/0d63ea6cd81b750abbafffc855a1d0ab> **<Kode3.js>**

Soal latihan, lengkapi kode berikut ini >> dan kirimkan jawabannya ke code playground >>
<http://jsbin.com/qomuxesavu/edit?html,js,output> **<Kode4.html>**. #belajarjs

same bagian fungsi dengan parsing argumen ada yang mau di diskusikan nggak
baiklah untuk mempersingkat waktu kita lanjut ke sub materi berikutnya yaitu

3) Fungsi Dengan Parsing Obyek

Seperti dibahas diawal, dalam javascript kita bisa membuat fungsi, dan saat pemanggilan fungsi tersebut, kita bisa memasukkan obyek sebagai argumennya .. #belajarjs

berikut contoh kode untuk fungsi dengan parsing obyek >> <https://jsfiddle.net/bvqr4qqp/> **<Kode5.html>**

lola ya, kalo ini >> <http://jsbin.com/vijitesafo/edit?html,js,output>
objek itu merupakan variable yang menampung nilai larik (array)

disitu ada fungsi cobaObyek dimana gunanya untuk mengambil nilai dari properti "merek"
dimana variabel motorKu digunakan untuk menyimpan nilai2 yang akan di gunakan pada fungsi cobaObyek()
sekian

ada yang mau nanya dulu nggak ? sebelum kita lanjutin ke pembahasan selanjutnya ? #belajarjs

aku aku ngirim jawaban latihan >> <http://jsbin.com/vudimamozi/1/edit?html,js,output> **<Kode6.html>**

4) Ekspresi Fungsi

Selain fungsi dideklarasikan dengan kata 'function' .. sebuah fungsi juga bisa dideklarasikan melalui sebuah ekspresi.. #belajarjs

contohnya pada kode berikut >> <http://jsbin.com/paxoqetowo/edit?html,js,output> **<Kode7.html>** #belajarjs

jadi var selain digunakan sebagai deklarasi variable, juga dapat digunakan untuk menampung data, sekaligus deklarasi fungsi (beserta hasil nya)

jadi itu ya yang dimaksud melalui ekspresi?
sepertinya iya

mungkin itu yang disebut .. mendeklarasikan fungsi melalui ekspresi..

While the function declaration above is syntactically a statement, functions can also be created by a function expression. Such a function can be anonymous; it does not have to have a name. For example, the function square could have been defined as:

contoh 2 >> <http://jsbin.com/baqihulozi/edit?html,js,output> <Kode8.html> #belajarjs

ga paham di a = map(multiply, [0, 1, 2, 5, 10]);
variable X sebelumnya tiada di berikan nilai

nah itu .. merenung dulu

kalau dicontoh2 sebelumnya biasanya multiply(nilai)
tapi ini hanya multiply
dan variable x didalam fungsi multiply tiada dikasi nila di awal, atau diberi nilai setelah dipanggil
bukankah multiply di angkut sebagai argumen map, di f?
terus di map ada perulangan yang mengkomputasi f
tapi diatas map(f, a), f itu sendiri tiada diberi nilai masukan
bahkan di a = map(multiply, [0, 1, 2, 5, 10]); hanya "mulltiply", yang umumnya fungsi variable masukan fungsi
didapat dari luar
multiply(masukan){}

tapi entah, coba Sandal lebih jelaskan

coba liat kode sederhana ini deh >> <https://jsfiddle.net/ov7jymLt/> <Kode9.html> #belajarjs

ini masih bisa dipahami, sama seperti fungsi refaktor
jika dilihat kan dapat nilai masukan dari luar

multiply sepertinya dimasukin .. ke argumen f .. pada fungsi map()

nah iya, x itu sendiri di dalam fungsi multiply apaa ? dapat nilai apa ?
masih belum dapat logikanya ayo dong

<https://www.sitepoint.com/higher-order-functions-javascript/>
di bagian "taking function as arguments"
yup, itu link bisa dipahami

hmm.. coba jelasin

terus di map ada perulangan yang mengkomputasi f .. << sepertinya dia benar.. sedangkan isi dari f sendiri kan .. dari fungsi multiply() .. sedangkan fungsi multiply() sendiri berisi.. perkalian dari sebuah nilai * sebuah nilai * sebuah nilai lagi ..

jadi pas argumen yang diparsingnya seperti ini >> multiply, [0, 1, 2, 5, 10] .. perulangan pertama 0 x 0 x 0 .. perulangan ke dua .. 1 x 1 x 1 .. perulangan ketiga .. 2 x 2 x 2 .. perulangan keempat .. 5 x 5 x 5 .. perulangan kelima 10 x 10 x 10 .. sehingga hasilnya ... 0, 1, 8, 125, 1000 *benar nggak yaa*

gimana kalo kita lanjut dulu.. kalo masih ada pertanyaan.. kita jadiin PR yang nggak terpecahkan #belajarjs
http://www.w3schools.com/js/js_function_parameters.asp

yang di parsing (di lewatkan) ialah argumen object, bla bla bla bla... seperti penjelasan teman2, (singkatnya) ok umumnya kita manggil fungsi dalam fungsi

PR >> kode latihan 5

di materi pada halaman MDN .. sub materi berikutnya adalah pemanggilan fungsi, karna pemanggilan fungsi dari tadi sudah kita bahas, kita lanjut ke sub materi berikutnya

5) Fungsi Rekursif

Berikut contoh kode latihannya >> <https://jsfiddle.net/7wewquk2/> <Kode10.html> #belajarjs

di contoh faktorial 10, kita memanggil fungsi faktorial dengan argumen nilai=10. di dalam faktorial dilakukan pemeriksaan nilai, karena sepuluh bukan nilai 0 atau satu maka dilakukan perkalian nilai 10 dengan faktorial 10-1 (9). dst, maaf, begitu benar?

sepertinya benar

yang lain ada yang mau ditanyain ga? sebelum lanjut knapa? alasannya? krn uda dijelaskan

contoh itu disebut rekursif, yaitu fungsi melakukan pemanggilan dirinya sendiri untuk melakukan perhitungan.. *benar nggak yaa*

6) Scope / Cakupan Fungsi

Dalam konteks bahasa pemrograman, scope atau cakupan merupakan sebuah aturan yang mengendalikan waktu hidup (lifetime) dan akses dari sebuah nilai dalam program. #belajarjs

Nilai yang dikendalikan oleh scope umumnya adalah variabel. Scope sangat penting dalam pemrograman, terutama untuk memudahkan kita dalam menamakan variabel dan manajemen memori. #belajarjs

Aturan scoping yang dimiliki Javascript hanyalah function scope. Function scope berarti semua parameter dan variabel yang dibuat di dalam sebuah fungsi tidak dapat diakses di luar fungsi tersebut #belajarjs

perhatikan kode berikut >> <http://jsbin.com/focosumoji/edit?html,js,output> <Kode11.html>

hmm.. ok .. ta jelasin dikit.. pada contoh diatas .. kita mendeklarasikan 3 buah variabel >> yaitu nilai1, nilai2 dan nama.. << variabel2 ini cakupannya global

artinya bisa diakses dari manasaja

sedangkan pada fungsi ambilSkor() .. kita memiliki 2 variabel.. yaitu nilai1, nilai2 .. kedua variabel ini cakupannya lokal.. hanya bisa diakses dari dalam fungsi ambilSkor() tersebut..
okeoke.lanjut pak

selain itu kita memiliki sebuah fungsi tambah() .. << fungsi tambah ini sendiri berada didalam fungsi ambilSkor() sehingga variabel yang akan dipakai adalah nilai2 dan nilai1 yang ada didalam fungsi ambilSkor() .. bukan yang dari variabel global.. ada yang tau ga alasannya kenapa ? #belajarjs
karena nested ga sih?

contoh 2 >> <http://jsbin.com/yiraziseco/edit?html,js,output> <Kode12.html>

cuma dibagian console.log(bill); diakan ngeprint di browser consol

sub materi berikutnya bisa dilihat di halaman MDN .. pas bagian >> Scope and the function stack

7) Scope And The Function Stack

kurang ngerti .. ini contoh kodenya >> <http://jsbin.com/bulazuvipa/edit?html,js,output> <Kode13.html>

foo(3); // coba nilai foo diganti pakai angka lain

yang lain tolong jelasin dong.. jangan diem aja.. nilai kesulitannya berkurang dibanding yang tadi parsing function in function parameter <<<

yang pasti .. fungsi tersebut foo() .. dipanggil dari dalam fungsi foo() sendiri

kalau diurut outputnya begin: 3 end: 0 begin: 2 end: 1 begin:1 end: 3

kita anggap pr
PR 2 >> kode latihan 8

didasar adalah yang disebut sebagai stack
stack - tumpukan

baiklah untuk mempersingkat waktu, marilah kita melanjutkan ke pembahasan selanjutnya #belajarjs

8) Fungsi Bersarang

Pada javascript kita bisa membuat fungsi didalam fungsi, hal ini disebut fungsi bersarang (nested function)

fungsi yang ada di dalam fungsi lainnya juga memiliki akses terhadap semua nilai-nilai yang dimiliki fungsi penampungnya #belajarjs

Ketika sebuah fungsi menggunakan dan bergantung kepada nilai yang ada di luar dirinya, fungsi tersebut dikenal dengan nama closure. #belajarjs

ini contoh kode latihannya >> <http://jsbin.com/bocecunawe/edit?html,js,output> <Kode14.html> #belajarjs

skarang kita meminta kesediannya kepada yang terhormat pak dion untuk menjelaskan maksud dari kode diatas ^ #belajarjs

kok hasilnya bisa 13, 25, 41 .. triawarman tolong jelasin

seperti sudah dijelaskan sebelumnya, fungsi didalam fungsi.
jadi di baris terakhir return dia memanggil fungsi diatasnya (masih dalam 1 blok fungsi), sekian

tambahPersegi(2,3) hasilnya 13 uih.. dapetnya dari menong?
krn $(2 \times 2) + (3 \times 3)$

ini contoh closure

<http://jsbin.com/ladiliqura/edit?html,js,output> <Kode15.html>

ada yang mau ditanyain ga? kalo ga kita lanjut ..

kode latihan 3 >> <http://jsbin.com/zirokujede/edit?html,js,output> <Kode16.html>

nah ini bagus dijadikan diskusi lagi
coba jelasin ka triawarman

hasilnya 8 sama 8 juga

kondisinya sama seperti dikusi awal tadi, yang dicari kemana nilai variable f berasal
ternyata variable f menampung objek, lebih tepatnya fungsi.

hmm.. trus?
kalau yang ini... bentar masih mencoba memahami

nah lagi2 ini di fungsi daleman(y), dapat nilai dari luaran(x)
didalamnya memproses $x + y$

ka ka dion ada tanggapan ga?

ternyata variable x dan y dlam fungsi daleman mendapat nilai dari luar...
contoh x didalam fungsi daleman mendapat masukan nilai dari luaran(x)

menurut ane .. ga tau bener apa nggak sih.. fungsi luaran() memiliki variabel x, dimana variabel tersebut bisa diakses dari mana saja dalam fungsi itu.. nah variabel itu juga dipakai buat nyimpen nilai .. *bener ga ya*
baru sadar y ga di parse saat return daleman(nilai);

ada yang mau tanggapin lagi gak?
variable x dapat masuka dari luaran(x), dan variable y seharusnya dapat masukan dari daleman(y), tetapi ternyata di code dibawahnya hanya "return daleman"
berharap ada return daleman(nilai);
butuh penjelasan lebih lanjut

ka matematika-adit sama ka greenshirt ga muncul2 ya..
aku bingung yang ada komentar think of it like bla bla
aku coba console.log(panggil_daleman); setelah baris itu
outputnya fungsi
nah, argumen 3 itu kemana?
melakukan perubahan kode

cara pemanggilan pertama itu memang harus 2 kali panggil gitu ya?
untuk memperoleh hasil yang diharapkan
kurang tau deh

jadi yang luaran(3) ternyata dia melewati fungsi lagi, bukan nilai
jika return daleman; diubah return daleman(x);

hasilnya?
hasilnya akan berbeda, dan hanya melakukan 1 alert saja.. aha

penjelasannya dong
pertanyaanya.. ini Sandal ga bermaksud menciptakan code bug kan ?
nggak .. nyontek dari halaman MDN ..

kasih penjelasan dong..
mudahnya anggap aja ini iterasi (penambahan).. i++;
beres, solved case.. walaupun ada logika yang ga sampai

untuk yang ini logikaku belum sampai
alert('panggil_daleman'+panggil_daleman);
coba dia memberikan return x + y;
*coba, dia memberikan return x + y;
3 diawal siapa yang nampung ?

atau bisa dianggap krn x didalam fungsi daleman ialah anak dari fungsi luaran, jadi dianya bisa langsung gunakan parameter masukan di induk

tapi saat hasil = panggil_daleman(5); kan umumnya 5 nimpa nilai 3

<http://jsbin.com/zirokujede/edit?html,js,output> <Kode16.html> kasus yang coba di cari logika berjalannya

aku menyadap x, console.log(x); didalam luaran sebelum daleman. ketika dia memanggil luaran(3). kenapa gak masuk ke x yang ada di daleman?

jadi panggil_daleman = luaran(3); dia kan memberikan nilai balik berupa fungsi
function daleman(y) { return x + y;}

lalu di panggil lagi oleh hasil = panggil_daleman(5);, disini nilainya 8
jadi logianya saat dipanggil pertama dia menyimpan nilai 3 pada x

lalu panggilan kedua krn dia memanggil fungsi sebelumnya dan diapat nlai balikan berupa fungsi function daleman(y) { return x + y;}, dan nilainya 5 pada y
jadi 3 + 5 = 8

bisa didapat ga ? atau penjelasannya masih rumit

mungkin itu sifat dari variabel pada javascript ya .. bisa menyimpan nilai, bisa menyimpan obyek.. bisa juga menyimpan fungsi..

panggil_daleman = luaran(3) << variabel panggil_daleman akan menyimpan fungsi ya..

tapi tapi tapi ...

<http://jsbin.com/lakiqogiwu/edit?js,output> <Kode17.html> modify dari sebelumnya

perhatikan di alert('panggil_daleman '+panggil_daleman); //function daleman(y) { return x + y;}

.. hmm.. ok karna pembahasan pada sub materi ini agak rumit .. Sandal mau nyeruput air putih dulu ..

baiklah karna waktunya sudah makin sedikit.. kita lanjutkan ke sub materi berikutnya..

9) Konflik Nama

Pada saat dua buah variabel .. yang beda cakupan (scope) memiliki nama yang sama .. akan terjadi konflik nama.. #belajarjs

cakupan variabel yang berada didalam fungsi akan lebih diprioritaskan .. dibandingkan variabel yang diluar fungsi #belajarjs #benergayaa

untuk lebih memperjelas.. bisa lihat kode berikkut >> <http://jsbin.com/dohasidowo/edit?html,js,console,output>
<Kode18.html> #belajarjs

kek nya bakal mandek disini lagi :(

curiga clue nya disini >> result = luaran0(20) << kalo kita panggil sebuah fungsi kan kek gini >> namafungsi() >> nah ini jadi gini >> namafungsi() << Sandal curiga kurung kedua digunain buat memparsing nilai kedalam inner function .. *bener ga yaa*

jadi result = luaran0(20) << memparsing nilai 20 ke dalam inner function

variabel x = 10 itu sebagai pancingan karena pokok permasalahannya adalah prioritas variabel yang sama. dan akhir dipilih variabel yang daa di dalam scope inner function.

karna outer function .. dan inner function di contoh kode diatas .. memiliki nama variabel yang sama .. sama2 'x' .. maka akan terjadi konflik nama .. pada konflik nama.. yang jadi prioritas adalah nama variabel yang ada didalam fungsi itu (inner) .. #belajarjs #cmiiw

kita lanjut pembahasannya ya?

penerapan web untuk input data, apakah bisa pakai javascript ?

apakah mungkin di web server yang very low end

apakah independent dari client web, misalnya dari android, pc, atau web robot, kayak google crawler segitu dulu

setahu ane bisa aja .. tapi semua kode javascript ajan dijalankan di browser.. #cmiiw..

kurang tau .. untuk browser di android .. apakah akan menjalankan versi javascriptnya sama atau tidak dengan web browser di pc..

sama kok, enginenya kebanyakan sama ow..

menurut ka presiden gimana?

kalo chrome/safari pake webkit, firefox pake gecko

kalo untuk jalanin di server, ada nodejs

iya itu..

contohnya botan dan mata-mata ini, dua bot ini dikode pake javascript. jalan di servernya gomix.com

node.js.. kalo ga salah.. web framework yang dibangun menggunakan javascript.. untuk lebih lanjut menanyakan tentang node.js bisa ditanyakan ke beliau>> presiden

baiklah kita lanjut ke sub materi berikutnya yaitu

10) Closures

Seperti yang sudah kita singgung didepan >> Ketika sebuah fungsi menggunakan dan bergantung kepada nilai yang ada di luar dirinya, fungsi tersebut dikenal dengan nama closure #belajarjs

bisa gak javascript kita pake di device lowend kayak arduino, raspberry-pi

colek pak presiden

ditampung aja pertanyaannya dulu, dilanjut om :)

ga apa2 sambil dijawab

mungkin pak engkur mau mengoprek tentang web server di perangkat lowend..

engkur: kalo raspberry-pi sih masih bisa. kalo arduino gak tau. terlalu berat buat jalanin runtimenya javascript keknya.

berikut adalah kode latihan closure >> <http://jsbin.com/gerokawuso/edit?html,js,output> <Kode19.html>
ada yang mau didiskusikan nggak dari kode diatas ^

yap ok, banyak baca baca tentang nodejs nih
apakah nodejs ini versi server dari javascript

gue mau buat IOT pakai arduino, raspbberry dan server beneran di internet atau di cloud
kayak begini lah

ada ini <http://johnny-five.io/>
sensor --> arduino --> (wifi-network) --> mini-pc (raspberry-pi) --> server (cloud server)

gak ada pertanyaan om :) lanjut deh :)

pada contoh kode diatas, variabel nama bisa diakses dari dalam fungsi2 lain .. dan tidak ada cara untuk mengakses variabel didalam fungsi2 lain .. kecuali dari dalam fungsi itu .. #belajarjs
baiklah kita lanjut ke sub materi berikutnya

11) Parsing Argumen Obyek Pada Fungsi

Pada javasript kita bisa memakai obyek sebagai nilai untuk dimasukan pada function #belajarjs

nilai yang di parsing sebagai argumen tersebut bisa kita akses seperti array.. #belajarjs

formatnya >> arguments[i] #belajarjs

perhatikan kode latihan berikut >> <http://jsbin.com/noritigure/edit?html,js,output> <Kode20.html> #belajarjs

ada yang mau didiskusiin nggak?

arguments[i] << dimana i dimulai dari index 0

hm... arguments itu kayak array, tapi bukan array?

ini >> <https://jsfiddle.net/e8ppk9q0/> <Kode20.html> << loading nggak

kayaknya mandek disini lagi :v

susah dipahamin..

baiklah kita lanjut ke sub materi berikutnya

12) Default Parameter Dan Rest Parameter

Default Parameter adalah jika salah satu argumen tidak ada, menggunakan nilai default #belajarjs

formatnya >> function namafungsi(defaultnilai = angka, defaultnilai2 = angka2) { // disini badan fungsinya }

Rest Parameter adalah menggunakan isi dari banyak nilai array sebagai aragumen..

contoh kode latihan untuk default parameter dan rest parameter >> <http://jsbin.com/qujiyapoli/edit?html,js,output> <Kode21.html> #belajarjs 'kode hanya berjalan di ecma script'

pada fungsi pertama pada baris >> function perkalian(a, b = 1) << defalut parameternya adalah 1

function perkalian(nilaipengali, ...argumen2) << pada contoh fungsi kedua .. nilai pertama dari awal.. digunakan sebagai pengali.. dan nilai2 selanjutnya sampe nilai terakhir dari array .. adalah angka2 yang akan dikalikan..

jadi array tersebut adalah >> [2, 1, 2, 3] << hasilnya adalah >> [2, 4, 6]

baiklah kita melanjutkan ke kode latihan berikutnya

13) Mempersingkat Fungsi

Perhatikan contoh latihan berikut >> <http://jsbin.com/qukaxafigu/edit?html,js,output> <Kode22.html>

pada kode di baris ini adalah sebuah fungsi panjang >> var a2 = a.map(function(s){ return s.length }); .. akan kita singkat jadi seperti ini >> var a3 = a.map(s => s.length); hasilnya sama aja :v

ok karna ga ada pertanyaan kita lanjut ke sub materi terakhir lexical this

14. Lexical This

Lexical this adalah sebuah variabel khusus dalam bahasa pemrograman oop yang digunakan sebagai penunjuk kepada objek, ketika kita mengaksesnya dari dalam class #belajarjs

berikut contohnya >> <http://jsbin.com/lixiyesive/edit?html,js,output> <Kode23.html> #belajarjs

Objek yang bagaimana yang ditunjuk om?

ini bisa dibuka nggak>> <https://jsfiddle.net/9bw3mn8s/> <Kode23.html>

hmm.. kalo dalam bahasa pemrograman php .. this digunakan ketika merujuk nilai sebagai obyek bukan sebagai class nya .. #belajarjs

kalo di php kan kita bikin class dulu.. nah class itu hanya sebagai cetakannya saja.. nanti kalo mau di pake di pemrograman.. kita harus menurunkannya menjadi obyek.. kita bisa menurunkan banyak obyek dari sebuah kelas.. #belajarjs

nah this ini akan merujuk kepada nilai didalam obyek2.. bukan didalam class.. maksudnya .. nilai2 this untuk tiap obyek belum tentu sama..

ok .. karna mengingat waktu terbatas.. maka kegiatan belajar bersama untuk hari ini pertemuan ke 3 saya akhiri sampe disini.. saya mengucapkan terima kasih untuk semua peserta yang telah hadir biarpun akhirnya bubar :v .. dan terima kasih kepada botan dan ChanServ yang sudah hadir dari awal same akhir kegiatan diskusi.. saya ucapkan wassalamu alaikum wr. wb.. happy learning \(_^)/

Catatan :

File PDF ini adalah log diskusi '**Belajar Bersama Javascript**' Yang diadakan di kanal irc **#ubuntu-indonesia**. Semua kode latihan bisa dilihat disini : <https://github.com/taufiqur-rahman/Belajar-Javascript-Dasar>