

ANALISIS TUGAS CODING OSPRODI 2020
PROGRAM STUDI S-1 INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET

Nama : Umar Hadi Pranoto

NIM : M0520078

Kelompok : JavaScript

1. Input/Output

Analisis Source Code

- Pada source code pertama-tama harus dideklarasikan constant *readline* dimana akan dioperator assignmentkan dengan tanda “=” dimana setelah tanda tersebut akan dipanggil operator `require('readline')` dimana `require` akan menjalankan dan cache modul-modul dalam JavaScript. Setelah itu perlu dilakukan pendeklarasian constant *membacaInputan* dimana akan dioperator assignmentkan dengan tanda “=” dimana setelah tanda tersebut akan dipanggil operator `readline.createInterface()` dimana `createInterface()` tersebut di dalamnya terdapat operasi `process.stdin` sebagai penginput dan `process.stdout` sebagai pengoutput.
- JavaScript mengambil satu baris input dari user menggunakan perintah `membacaInputan.question("", function(variabel){})` dimana fungsi `membacaInputan` yang telah dibuat akan dipanggil terlebih dahulu dan di dalam fungsi `question()` akan dijalankan berbagai operasi untuk fungsi, pendeklarasian, ataupun pernyataan didalam operator `function()` didalam operator `question()`.
- Pada soal input dan output, mengacu kepada source code yang telah dibuat, inputan dibuat dengan memanggil fungsi `membacaInputan.question("", function(variabel))` dimana variabel berturut-turut dideklarasikan dengan isi: nama lengkap, nim, umur, asal daerah, dan hobi dan kemudian disimpan ke variabel masing-masing (nama, nim, umur, asal, hobi).
- Kemudian baris-baris berikutnya menampilkan output sesuai format yang telah ditentukan, menggunakan fungsi `console.log` yang mana berfungsi untuk menampilkan output Cara menggabungkan variabel yang telah diinput untuk ditampilkan ke output yaitu yaitu hanya dengan memanggil variabel yang dinginka yang diletakkan bersama teks dengan operator + dimana operator tersebut akan menggabungkan teks dengan pemanggilan variabel pada

perintah fungsi `console.log`. Program ini akan diakhiri dengan `process.exit()` dimana program akan berakhir.

Analisis Program

- Program dijalankan menggunakan input yaitu:

Umar Hadi Pranoto

M0520078

18

Karanganyar

Bermain game

Kemudian program menampilkan output berupa:

Halo, namaku Umar Hadi Pranoto dengan NIM M0520078.

Sekarang aku berumur 18 tahun dan berasal dari Karanganyar.

Hobiku adalah Bermain game.

Salam kenal semuanya!

- Program awalnya menerima input berupa 5 baris berturut-turut yang berisi data di atas.

Kemudian program memasukkan masing-masing baris input ke variabel yang bersangkutan:

Umar Hadi Pranoto → *nama*

M0520078 → *nim*

18 → *umur*

Karanganyar → *asal*

Bermain game → *hobi*

- Kemudian setelah masing-masing input disimpan pada variabel, variabel-variabel tersebut digunakan untuk mensubstitusikan variabel yang terdapat pada format output, sehingga jika menulis nama akan menghasilkan teks “Umar Hadi Pranoto”, dan seterusnya untuk semua variabel. Sebagai contoh, jika kita menuliskan `console.log("Hobiku adalah " + hobi + ".");`, hobi pada perintah `console.log` akan disubstitusi oleh variabel *hobi*, dalam kasus tersebut, variabel *hobi* berisi teks “Bermain game” sehingga transformasinya menjadi “*Hobiku adalah Bermain game.*”

2. Pembagian Kelas

Analisis Source Code

- Pada source code pertama-tama harus dideklarasikan constant *readline* dimana akan dioperator assignmentkan dengan tanda “=” dimana setelah tanda tersebut akan dipanggil operator `require('readline')` dimana `require` akan menjalankan dan cache modul-modul dalam JavaScript. Setelah itu perlu dilakukan pendeklarasian constant *membacaInputan* dimana akan dioperator assignmentkan dengan tanda “=” dimana setelah tanda tersebut akan dipanggil operator `readline.createInterface()` dimana `createInterface()` tersebut di dalamnya terdapat operasi `process.stdin` sebagai penginput dan `process.stdout` sebagai pengoutput.
- JavaScript mengambil satu baris input dari user menggunakan perintah `membacaInputan.question("", function(variabel){})`) dimana fungsi `membacaInputan` yang telah dibuat akan dipanggil terlebih dahulu dan di dalam fungsi `question()` akan dijalankan berbagai operasi untuk fungsi, pendeklarasian, ataupun pernyataan didalam operator `function()` didalam operator `question()`.
- Pada soal pembagian kelas, mengacu pada source code yang telah dibuat, menginstruksikan program untuk membaca input dari user dengan fungsi `membacaInputan.question("", function(variabel))`. fungsi `membacaInputan.question("", function(variabel))` masing-masing dideklarasikan secara berturut-turut dengan variabel yaitu nama dan nim dimana inputan masing-masing disimpan pada variabel *nama* dan *nim*.
- Pengecekan kelas A, B, atau C menggunakan fungsi `if else`. `If` akan dijalankan jika kondisi atau pernyataan yang dideklarasikan bernilai 1 atau *true* atau benar. `Else` akan dijalankan jika kondisi atau pernyataan yang dideklarasikan bernilai 0 atau *false* atau salah. Terdapat penggunaan `else if` jika kondisi yang diinginkan lebih dari satu.
- Pada suatu kondisi dapat dijalankan suatu pernyataan logika salah satunya adalah “&&” dimana kondisi atau pernyataan di kanan atau pun kiri tanda tersebut kondisinya harus dipenuhi.
- Pengecekan kelas A, B, atau C menggunakan fungsi `if else` dimana didapat dengan tiga kondisi. Kondisi yang pertama apabila variabel *nim* bernilai `nim >= 1` dan `nim < 30` maka program akan mensubstitusi nilai variabel *kelasABC* dengan string “A”. Kondisi kedua apabila *nim* bernilai `nim >= 30 && nim < 58` maka program akan mensubstitusi nilai variabel *kelasABC* dengan string “B”. Kondisi ketiga `nim >= 58` maka program akan mensubstitusi nilai variabel *kelasABC* dengan string “C”.

- Pengecekan ganjil genap menggunakan operasi modulo atau sisa pembagian sehingga program mengecek sisa pembagian jika suatu variabel x dibagi 2. Sisa pembagian dari 2 hanya salah satu dari antara 0 dan 1, di mana jika bersisa 0 atau habis dibagi 2 maka variabel x tersebut merupakan bilangan bulat genap, dan jika bersisa 1 atau tidak habis dibagi 2 maka variabel x tersebut merupakan bilangan bulat ganjil.
- Program melakukan pengecekan apakah variabel *nim* (yang mana berisi 2 digit terakhir pada NIM) tidak habis dibagi 2 dengan melakukan pengecekan: *if nim % 2 == 1*. Jika kondisi tersebut memenuhi (dalam arti *nim* tidak habis dibagi 2), program akan mensubstitusi nilai variabel *ganjilGenap* dengan string “genap”. Namun jika kondisi tersebut tidak memenuhi (dalam arti *num* habis dibagi 2), program akan mensubstitusi nilai variabel *ganjilGenap* dengan string “ganjil”.
- Program akan menampilkan seluruh output dengan fungsi 2 baris berturut-turut yaitu `console.log(`Halo, namaku ${nama}.`); console.log("Pada semester ini aku mendapat kelas " + kelasABC + " dan " + ganjilGenap + ".");`. Program ini akan diakhiri dengan `process.exit()` dimana program akan berakhir.

Analisis Program

- Program dijalankan menggunakan input yaitu:

Umar Hadi Pranoto

78

Kemudian program menampilkan output berupa:

Halo, namaku Umar Hadi Pranoto.

Pada semester ini aku mendapat kelas C dan genap.

- Setiap baris pada input masing-masing dimasukkan ke dalam variabel yang bersangkutan:

Umar Hadi Pranoto → *nama*

78 → *nim*

- Pada awalnya, program menampilkan satu baris output berupa “*Halo, namaku Umar Hadi Pranoto.*” menggunakan `console.log` di mana format *nama* pada `console.log` disubstitusi dengan teks yang terdapat pada variabel *nama* (dalam kasus tersebut “*Umar Hadi Pranoto*”)

- Kemudian program menentukan bahwa *nim* (yang diinput dengan angka 78) masuk pada kelas C dan genap. Hal tersebut dikarenakan 78 berada pada rentang kelas C yaitu $nim \geq 58$ dan 78 merupakan bilangan genap ($78 = 2 \cdot 39 + 0$ atau $48 \pmod{2} = 0$). Sehingga program mengeluarkan output “*Pada semester ini aku mendapat kelas C dan genap.*”

3. Generate NIM

Analisis Source Code

- Pada source code pertama-tama harus dideklarasikan constant *readline* dimana akan dioperator assignmentkan dengan tanda “=” dimana setelah tanda tersebut akan dipanggil operator `require('readline')` dimana `require` akan menjalankan dan cache modul-modul dalam JavaScript. Setelah itu perlu dilakukan pendeklarasian constant *membacaInputan* dimana akan dioperator assignmentkan dengan tanda “=” dimana setelah tanda tersebut akan dipanggil operator `readline.createInterface()` dimana `createInterface()` tersebut di dalamnya terdapat operasi `process.stdin` sebagai penginput dan `process.stdout` sebagai pengoutput.
- JavaScript mengambil satu baris input dari user menggunakan perintah `membacaInputan.question("", function(variabel){})` dimana fungsi *membacaInputan* yang telah dibuat akan dipanggil terlebih dahulu dan di dalam fungsi `question()` akan dijalankan berbagai operasi untuk fungsi, pendeklarasian, ataupun pernyataan didalam operator `function()` didalam operator `question()`.
- Pada soal pembagian kelas, mengacu pada source code yang telah dibuat, menginstruksikan program untuk membaca input dari user dengan fungsi `membacaInputan.question("", function(variabel))`. fungsi `membacaInputan.question("", function(variabel))` masing-masing dideklarasikan secara berturut-turut dengan variabel yaitu *tahun* dan *banyak* dimana inputan masing-masing disimpan pada variabel *tahun* dan *banyak*.
- Pengecekan digit dalam variabel *tahun* dilakukan dengan `if else` dan menghasilkan 2 kondisi. Sebelumnya dilakukan pendeklarasian variabel *digitTahun*. Kondisi yang pertama apabila `tahun.length < 2` dimana `length` menyatakan banyaknya digit tahun maka program akan mensubstitusi nilai variabel *digitTahun* dengan `digitTahun = "0" + tahun`. Kondisi yang kedua apabila `tahun.length = 2` dimana `length` menyatakan banyaknya digit tahun maka program akan mensubstitusi nilai variabel *digitTahun* dengan `digitTahun = tahun`.

- Kemudian setelah menerima input, pada baris berikutnya program memasuki tahap looping dari 1 sampai dengan *banyak*. Hal ini berarti looping atau pengulangan dilakukan sebanyak *banyak* kali atau sesuai dengan baris kedua input dari user.
- Looping berfungsi untuk mengeksekusi rangkaian perintah yang sama secara berulang-ulang dengan memanfaatkan perubahan variabel pada loop, jadi dapat dimodelkan sebagai perubahan eksekusi rangkaian perintah terhadap variabel pada loop. Dalam kasus ini, variabel loop yang digunakan adalah *i* yang menyatakan nomor urut mahasiswa (yang memiliki rentang [1, banyak])
- Sintaks for loop pada JavaScript adalah: `for(i = 1; i <= banyak; i++)` untuk mengulang-ulang perintah seiring berjalannya variabel dari A sampai B secara terurut (misal `for(i = 1; i <= 10; i++)` mengulang perintah sebanyak 10 kali untuk *x* dari 1 sampai 10 secara urut).
- Sehingga penulisan di dalam JavaScript adalah: `for(i = 1; i <= banyak; i++)` di mana perintah diulang sebanyak *banyak* kali (yaitu jumlah mahasiswa), bergantung dengan input yang dimasukkan oleh user (baris kedua dalam inputan program).
- Perintah yang dieksekusi di dalam loop adalah `for(i = 1; i <= banyak; i++)`. Hal tersebut berarti output yang ditampilkan selalu berubah seiring berubahnya nilai *i*. Pengecekan nilai bernilai dengan beberapa kondisi menggunakan fungsi if else dimana didapat dengan tiga kondisi. Kondisi yang pertama apabila variabel *i* bernilai `nim < 10` maka program akan mensubstitusi nilai variabel *nim* dengan `nim = "00" + i`. Kondisi kedua apabila variabel *i* bernilai `nim < 100` maka program akan mensubstitusi nilai variabel *nim* dengan `nim = "0" + i`. Kondisi ketiga apabila variabel *i* bernilai `nim < 1000` maka program akan mensubstitusi nilai variabel *nim* dengan `nim = i`.
- Perintah output dilakukan dengan operasi `console.log` dengan satu baris yaitu `console.log("M05" + digitTahun + nim)`. Program ini akan diakhiri dengan `process.exit()` dimana program akan berakhir.

Analisis Program

- Program dijalankan menggunakan input yaitu:

10

Kemudian program menampilkan output berupa:

M0520001

M0520002

M0520003

M0520004

M0520005

M0520006

M0520007

M0520008

M0520009

M0520010

- Program awalnya menerima input berupa 2 baris berturut-turut yang berisi data di atas (2 digit tahun angkatan (20) pada baris pertama dan jumlah mahasiswa (10) pada baris kedua), kemudian masing-masing dimasukkan dalam variabel:

20 → *tahun*

10 → *banyak*

- Program ini berarti meng-generate NIM untuk mahasiswa informatika (kode prodi 05) fakultas MIPA (kode fakultas M) angkatan 2020 sebanyak 10 kali (diasumsikan mahasiswa informatika FMIPA angkatan 2020 adalah 10), maka NIM yang tergenerate adalah seperti pada output di atas (mulai dari M0520001 sampai dengan M0524010).
- Dengan memanfaatkan looping, dapat diperhatikan bahwa 3 digit terakhir pada NIM pada tiap baris selalu berubah seiring berjalannya loop, mengikuti perubahan variabel pada loop (dalam kasus tersebut adalah *i* yang terletak pada range [1, banyak] atau [1, 10] karena variabel *banyak* bernilai 10), di mana perintah yang terdapat di dalam loop adalah perintah if else yang akan memilah data *i*. Perintah output akan dilakukan dengan operator *console.log("M05" + digitTahun + nim)*. Kemudian *console.log* dengan format tersebut dalam loop menampilkan satu *M0524XXX* di mana XXX adalah tiga digit nomor urut (n). Perintah tersebut diulang sebanyak 10 kali dengan nilai *i* yang berbeda (dari 1 sampai 10 atau sebanyak *banyak*) sehingga dapat meng-generate seluruh NIM dari M0524001 sampai dengan M0524010.