**MATERI PERTEMUAN 1**

# Notasi Algoritma Dalam Pemrograman

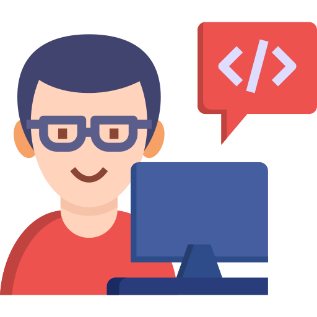
**TAHUKAH KAMU…?**

“Cara penulisan Algoritma?”

“Bagaimana Penulisan Kalimat Deskriptif?”

“Bagaimana Pseudocode?”

“Bagaimana Notasi Flowchart?”



Algoritma dapat didefinisikan sebagai kumpulan instruksi yang berurutan secara sistematis, yang menggambarkan langkah – langkah yang perlu diikuti untuk menyelesaikan permasalahan tertentu dengan bantuan komputer. Instruksi dalam algoritma merupakan perintah untuk melakukan aksi tertentu, misalnya instruksi menulis hasil perhitungan, membaca masukan, memanggil sebuah prosedur, dan sebagainya. Instruksi algoritma dinyatakan sebagai pernyataan yang dapat disajikan dalam bentuk tulisan. Tulisan pernyataan instruksi algoritma haruslah menggunakan bahasa yang mudah dibaca dan dipahami oleh manusia. Penyajian penulisan algoritma ini dapat dilakukan menggunakan untaian kalimat deskriptif, pseudocode dan flowchart. Bab 3 ini akan menjelaskan tentang bagaimana penggunaan dan penyajian notasi algoritma pemrograman dalam bentuk kalimat deskriptif, pseudocode, dan flowchart. Selain itu, di bab ini juga akan menjelaskan beberapa contoh penerapan notasi algoritma pada beragam kasus bisnis.

1. **Penulisan Kalimat Deskriptif**

Pada dasarnya, sebuah algoritma merupakan langkah – langkah sistematis yang dapat digunakan untuk penyelesaian permasalahan tertentu. Penulisan algoritma dapat disajikan dalam bentuk kalimat deskriptif. Penulisan notasi algoritma menggunakan kalimat deskriptif disebut dengan notasi alami.

Penulisan instruksi – instruksi dalam notasi algoritma deskriptif dilakukan dalam bentuk kalimat deskriptif dengan menggunakan bahasa yang jelas. Notasi algoritma deskriptif ini biasanya digunakan pada algoritma yang pendek karena apabila untuk algoritma yang panjang notasi deskriptif kurang efektif. Tidak aturan yang menuliskan algoritma deskriptif. Namun, penulisan algoritma menggunakan kalimat deskriptif ini dapat menimbulkan ambiguitas, dimana kalimat dapat mengandung makna ganda sehingga dapat terjadi kesalahpahaman saat membaca algoritma tersebut. Algoritma yang pendek akan lebih efektif jika ditulis menggunakan untaian kalimat deskriptif ini.

Algoritma dengan notasi kalimat deskriptif biasanya menggunakan berbagai kata kerja dalam bahasa manusia, seperti mulai, tulis, baca, tampilkan, jika, maka, ulangi. Adapun menggunakan bahasa manusia, namun dalam Bahasa Inggris, seperti read, print, write, if, end, dsb. Tidak ada ketentuan didalam penulisan notasi kalimat deskriptif, penulis bebas dalam menuliskan notasi, yang penting tulisan tersebut jelas dan mudah dimengerti oleh pembaca. Dalam penulisannya, notasi kalimat deskriptif terbagi menjadi 3 bagian, antara lain: **judul, deklarasi,** dan **deskripsi**. Bagian judul merupakan bagian yang terdiri dari nama algoritma dan penjelasan tentang algoritma yang akan dibuat. Bagian deklarasi mendefinisikan semua nama yang digunakan pada algoritma, seperti variable, tipe data, fungsi, konstanta. Bagian deskripsi mendefinisikan uraian langkah – langkah penyelesaian masalah.

Adapun contoh dari penulisan notasi kalimat deskriptif adalah sebagai berikut:

**Contoh 1:**

Judul:

Algoritma\_Perkalian

{perkalian 2 bilangan bulat. Algoritma memasukkan 2 bilangan bulat, kemudian menghitung hasil perkalian dari kedua angka tersebut, dan mencetak hasil perkalian tersebut}

Deklarasi:

bi1, bi2, hasilKali = int Deskripsi:

Input Bi1, Bi2

hitung hasilKali = bi1 x bi2 Tampilkan hasilKali

Selesai

**Contoh 2:**

Judul: Algoritma\_hitungDiskon

{menghitung harga setelah diskon dengan memasukkan harga awal produk dan besaran diskonnya}

Deklarasi:

hargaAwal, besaranDiskon(%), hargaDiskon = int Deskripsi:

Input hargaAwal, besaranDiskon(%)

Hitung hargaDiskon = hargaAwal-(hargaAwal x besaranDiskon(%))

Tampilkan hargaDiskon

Selesai.

Kedua contoh algoritma diatas dapat menunjukkan bahwa penggunaan notasi kalimat deskriptif sangatlah sederhana, mudah dibaca dan dipahami. Penyelesaian permasalahan yang sederhana lebih mudah menggunakan notasi kalimat deskriptif ini, namun jika penyelesaian masalah terbilang kompleks dan rumit, akan lebih sulit terselesaikan menggunakan notasi kalimat deskriptif, dan dapat menggunakan notasi algoritma lainnya, seperti pseudocode atau flowchart.

1. **Pseudocode**

Algoritma dapat ditulis menggunakan notasi pseudocode. Penulisan menggunakan notasi pseudocode ini memerlukan penjabaran yang selaras dengan logika, sehingga menjadikan algoritma mudah dibaca dan dipahami. Pseudocode berasal dari kata pseudo yang berarti imitasi atau mirip atau menyerupai dan code yang berarti kode program. Dengan kata lain, pseudcode dalam sebuah algoritma bukanlah kode program sebenarnya, hanya mirip saja, sehingga tidak dapat langsung dieksekusi oleh komputer. Pseudocode harus melalui proses translasi terlebih dulu. Translasi merupakan proses penerjemahan pseudocode ke dalam bahasa pemrograman tertentu. Terdapat berbagai macam bahasa pemrogaraman yang dapat dieksekusi oleh komputer, seperti Bahasa Java, Bahasa Pascal, Bahasa C++, dan sebagainya. Sama halnya dengan kalimat deskriptif, penulisan algoritma menggunakan pseudocode juga terbilang mudah. Secara umum, tidak ada aturan atau notasi pseudocode yang bdiuntuk menuliskan algoritma. Notasi yang digunakan juga menggunakan bahasa manusia, seperti read, print, write, if, end.

Hal ini berbeda dengan notasi bahasa pemrograman yang harus mematuhi aturan– aturan tertentu agar program dapat berjalan. Proses penerjemahan atau translasi dari pseudocode ke kode program mudah dilakukan karena terdapat korespondensi antara notasi pseudocode dan notasi bahasa pemrograman atau dengan kata lain, sebelum mengenal bahasa pemrograman, sebaiknya kita mengetahui terlebih dulu bagaimana alur alogaritma suatu program menggunakan pseudocode.

1. **Pentingnya Menggunakan Pseudocode**

Pada dasarnya, pseudocode merupakan notasi algoritma yang dapat dijadikan acuan untuk membuat sebuah program. Pseudocode ini bukanlah kode pemrograman sebenarnya, namun penulisannya menggunakan bahasa tiruan dari bahasa pemrograman. Pseudocode dapat memecahkan permasalahan dengan kalimat sederhana, sehingga dapat mudah dibaca dan dipahami oleh manusia. Sedangkan, bahasa pemrograman merupakan bahasa yang nantinya akan dibaca oleh software dalam pembuatan program. Berikut adalah beberapa uraian tentang pentingnya menggunakan pseudocode:

* **Memberikan kemudahan dalam pemahaman**

Konsep dan struktur pseudocode dapat dipahami dengan mudah, sehingga akan memberikan kemudahan dalam membuat rancangan alur algoritma yang benar dan tepat. Pseudocode dapat memberikan pemahaman alogaritma yang baik sebelum mempelajari bahasa pemrograman yang lebih kompleks.

* **Menjadi sarana penghubung yang baik**

Pseudocode merupakan notasi alogaritma yang memberikan kemudahan bagi para pemula yang sedang ingin merancang sebuah progam. Pseudocode mampu menjadi penghubung yang baik, yang mana mampu menyambungkan logika dengan program yang sedangan dirancang.

* **Dapat digunakan sebagai dokumentasi.**

Penulisan notasi algoritma menggunakan pseudocode dapat menjadi sarana dokumentasi. Hal ini dikarenakan pseudocode yang sudah dibuat dapat disimpan dan dapat dikembangkan lagi apabila akan melakukan peningkatan pada alur alogaritma pemrograman.

1. **Penerjemah flowchart.**

* Notasi Pseudocode dapat digunakan oleh programmer pemula untuk menerjemahkan flowchart ke kode pemrograman dengan lebih efisien dan lebih efektif.
* Memudahkan Proses Pengembangan Sistem
* Pseudocode dapat membantu dalam proses pengembangan aplikasi, karena menggunakan bahasa sederhana dan struktur sederhana sehingga mudah dibaca dan mudah dimodifikasi.

1. **Ciri – ciri Pseudocode**

* Pseudocode menggunakan bahasa yang tidak rumit, yang mana mudah dibaca dan dipahami oleh manusia.
* Pseudocode tidak memiliki aturan atau standar bdiyang harus diikuti oleh pembuatnya
* Pseudocode menggunakan sintaks dari suatu program, seperti ←, <, >, <=,
* >=, dll
* Pseudocode ditulis berupa langkah – langkah yang berurutan untuk menyelesaikan permasalahan.

1. **Notasi Pseudocode**

Penggunaan notasi pada pseudocode memiliki tujuan untuk memahami fungsi-fungsi dan algoritma di sebuah kode pemrograman. Pada pembahasan sebelumnya, telah dipahami bahwa pseudocode tidak memiliki aturan yang bdidalam penggunaan notasi. Namun, pada umumnya pseudocode menggunakan beberapa notasi sebagai berikut:

* ***Input***: notasi ini digunakan untuk memasukkan isi, seperti klik atau tombol. Contoh notasi untuk *Input* adalah Read, Get, Obtain.
* ***Output***: notasi ini digunakan untuk menampilkan hasil dari *Input* dan proses. Contoh notasi untuk *Output* adalah Print, Write, Display, Show.
* ***Compute***: notasi ini digunakan untuk memproses atau menghitung masukkan. Contoh notasi untuk *Compute* adalah Compute, Calculate, Determine.
* ***Decision***: notasi ini digunakan untuk menunjukkan adanya percabangan pada algoritma. Contoh notasi untuk *Decision* adalah If-then-else, Case.
* ***Iteration***: notasi ini digunakan untuk menunjukkan adanya perulangan pada algoritma. Contoh notasi untuk *Iteration* adalah While, For, Repeat-Until. Notasi While digunakan untuk sebuah perulangan yang memiliki iterasi awal. For digunakan untuk sebuah perulangan perhitungan iterasi. Sedangkan, Repeat-Until digunakan sebuah perulangan yang memiliki kondisi akhir.

1. **Struktur pseudocode**

* **Judul**

Bagian judul berisi nama program yang akan dibuat. Judul dapat menyatakan penjelasan singkat tentang program yang akan dibuat. Biasanya bagian ini penulisannya diawali dengan kata “PROGRAM”, kemudian diikuti dengan nama programnya. Umumnya, nama program terdiri dari satu kata, singkat, namun menggambarkan apa yang dilakukan oleh program. Setelah menyatakan nama program, maka dapat diberikan uraian singkat yang mendeskripsikan program dibawah nama program. Berikut adalah contoh penulisannya:

PROGRAM LuasLingkaran

{Program membaca Panjang jari-jari (r) sebuah lingkaran,

menghitung luas lingkaran, lalu mencetak luas tersebut ke layar.}

* **Deklarasi**

Bagian deklarasi digunakan untuk menyatakan semua nama yang digunakan di dalam algoritma. Nama yang dideklarasikan dapat berupa nama konstanta, nama peubah, nama tipe data, nama prosedur atau fungsi. Dalam algoritma, semua nama harus dikenali terlebih dulu sebelum mereka digunakan. Bagian deklarasi ini dapat dikosongkan apabila tidak ada penggunaan nama didalam bagian isi algoritma. Berikut adalah contoh penulisannya:

DEKLARASI:

const pi=3.14 {konstantaπ}

r: real {jari – jari lingkaran dalam satuan cm}

L: real {luas lingkaran dalam satuan 〖cm〗^2

* **Isi Algoritma**

Bagian isi merupakan bagian utama dari algoritma, berisi sekumpulan instruksi atau langkah – langkah yang sistemastis, kondisional, atau berulang untuk menjalankan program dengan menggunakan nama – nama yang telah dideklarasikan pada bagian deklarasi. Berikut adalah contoh penulisannya:

ALGORITMA:

read(r) {masukkan jari-jari lingkaran}

L ← pi\*r\*r {hitung luas lingkaran}

write(L) {tampilkan luas lingkaran ke layar}.

1. **Penggunaan notasi Pseudocode**

Tujuan utama dari dibuatnya pseudocode adalah agar manusia dapat membaca atau menyelesaikan permasalahan dengan sebuah algoritma. Pseudocode berisi instruksi – instruksi sistematis yang berguna untuk menyelesaikan suatu permasalahan.

Seorang programmer membutuhkan pseudocode untuk digunakan sebagai dasar alogaritma pengembangan sebuah program. Untuk membuat sebuah program, pseudocode diubah / ditranslasi ke bahasa pemrograman. Terdapat berbagai bahasa pemrograman yang dapat digunakan untuk mengembangkan suatu program, seperti Bahasa Java, Bahasa Pascal, Bahasa C++, dan sebagainya. Untuk pembahasan lebih lanjut, bahasa pemrograman yang akan digunakan pada buku ini hanyalah bahasa Java saja. Seperti pada pembahasan sebelumnya, dapat diketahui bahwa penggunaan notasi pseudocode sangatlah mudah, karena tidak ada aturan yang perlu diikuti. Berikut ini diuraikan contoh – contoh penggunaan notasi pseudocode:

**Contoh 1**

Instruksi (pernyataan): Tulis kalimat “Semangat belajar, ya!”

Maka, notasi pseudocodenya dapat ditulis: write (Semangat belajar, ya!)

Notasi pseudocode ditulis dengan sederhana, tidak memikirkan bagaimana tampilan dari luaran yang diinginkan. Sebaliknya, jika dalam bahasa pemrograman, terdapat instruksi yang dapat digunakan untuk menampilkan luaran dengan berbagai format. Misalnya, “Semangat belajar, ya!” ingin ditampilkan dalam satu baris atau antara “semangat belajar” dan “ya!” ditampilkan menggunakan koma atau hanya spasi saja. Tampilan luaran tersebut dapat diperoleh dengan menggunakan aturan aturan yang berldididalam bahasa pemrograman.

Notasi Write berfungsi untuk mencetak kalimat “Semangat belajar, ya!”, Notasi Write di dalam algoritma berkoresponden dengan print dan println dalam bahasa Java. Pada bahasa pemrograman Java antara setiap instruksi dipisahkan dengan tanda “;” (semicolon).

Dengan demikian, translasi Write (Semangat belajar, ya!) ke bahasa Java adalah:

Notasi pseudocode: write (Semangat belajar, ya!)

Notasi Bahasa Java: System.out.print (“Semangat belajar, ya!”);

**Contoh 2**

Instruksi (pernyataan): Memasukkan nilai m ke dalam n

Maka, notasi pseudocodenya dapat ditulis: **n←m**

Notasi **"←"** pada pseudocode memiliki arti yaitu mengisikan nilai m ke dalam variable n. Translasi notasi **"←"** dalam bahas Java adalah **"=".**

Sehingga,

**n←m** ditranslansi ke Bahasa Java menjadi: **n=m**

1. **Notasi Flowchart**

Flowchart merupakan standar dasar pemodelan yang berfungsi untuk menggambarkan alur logika dalam algoritma pemrograman. Flowchart ditampilkan dalam bentuk diagram yang mana dapat merepresentasikan urutan dari operasi yang dilakukan dari awal sampai akhir untuk penyelesaian algoritma pemrograman, sehingga mendapatkan hasil. Penggunaan flowchart akan membuat algoritma pemrograman menjadi lebih mudah dimengerti, karena algoritma ditampilkan dalam simbol – simbol gambar yang terstandar. Tujuan penggunaan flowchart adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, ringkas dan jelas dengan menggunakan simbol – simbol yang mudah dipahami.

1. **Jenis – jenis Flowchart**

* **Flowchart sistem** merupakan diagram alir yang menggambarkan arus pekerjaan secara keseluruhan dari suatu sistem. Flowchart sistem menjabarkan urutan – urutan dari prosedur – prosedur yang ada didalam sistem.
* **Flowchart dokumen** merupakan diagram alir yang menggambarkan arus dari laporan atau formulir.
* **Flowchart skematik** merupakan diagram alir yang hampir sama dengan flowchart sistem, yaitu menggambarkan arus prosedur yang ada didalam sistem. Perbedaanya adalah dalam pembuatannya, flowchart skematik tidak hanya menggunakan simbol namun juga menggunakan gambar-gambar computer dan peralatan lainnya yang digunakan. Gambar digunakan untuk memudahkan komunikasi kepada orang yang kurang paham terkait diagram alir yang dibuat.
* **Flowchart program** merupakan diagram alir yang menggambarkan urutan atau langkah – langkah logika dari suatu prosedur pemecahan masalah. Untuk flowchart program ini dapat dibuat dengan menggunakan metode penggambaran flowchart sebagai berikut:
* **Conceptual flowchart**, menggambarkan alur pemecahan masalah secara global
* **Detail flowchart**, menggambarkan alur pemecahan masalah secara rinci.
* **Flowchart proses** merupakan diagram alir yang menggambarkan rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah dalam suatu prosedur atau sistem.

1. **Pentingnya Penggunaan Flowchart**

Flowchart memiliki peranan penting bagi perusahaan di berbagai bidang. Keberadaan flowchart digunakan oleh perusahaan sebagai acuan untuk menjalankan suatu proses atau aktivitas bisnis. Flowchart juga memiliki fungsi yang cukup penting didalam proses pengembangan sebuah program. Penggunaan flowchart menjadi sangat penting karena beberapa alasan sebagai berikut:

* **Flowchart dapat memperjelas proses yang kompleks**

Sebuah program yang rumit dan kompleks pasti sulit untuk dipahami. Pengembang program membutuhkan flowchart untuk berkomunikasi dengan pelanggannya. Pelanggan merupakan orang awam yang perlu dijelaskan tentang algoritma dari berjalannya sebuah program. Dengan flowchart, maka kerumitan akan menjadi lebih sederhana, pelanggan lebih mudah menerima penjelasan yang disampaikan.

* **Flowchart dapat meningkatkan efisiensi**

Flowchart dapat digunakan untuk meninjau ulang alur kerja sebuah program. Sebuah program masih dapat diperbaiki atau dioptimalkan dengan penyederhanaan proses maupun pengalihan alur melalui flowchart. Dengan demikian, flowchart dapat memberikan efisiensi dalam memperbaiki dan mengoptimalkan sebuah program.

Flowchart dapat mengetahui proses pada program yang dijalankan Dengan menggunakan flowchart, pengembang atau pelanggan dapat melihat bagaimana sebuah proses berjalan ketika program mulai diaktifkan.Hal ini dikarenakan simbol – simbol flowchart yang dapat memberikan keterangan secara jelas dan detil mengenai apa yang terjadi didalam sebuah program. Misalnya, flowchart untuk perhitungan harga diskon, maka akan terdapat simbol input yang berisi harga awal, besaran diskon dan juga ada simbol proses yang digunakan untuk melakukan perhitungan harga diskon, serta terdapat simbol output untuk menampilkan harga diskon.

* **Flowchart dapat digunakan sebagai acuan proses pengembangan**

Sebelum melakukan koding pada program, para tim pengembang membuat flowchart terlebih dulu. Notasi flowchart digunakan dalam proses perancangan sebuah program. Dalam penerapannya, flowchart ini digunakan sebagai dasar atau acuan untuk melakukan koding, sehingga pengembang akan lebih mudah memahami alur alogaritma dari program yang akan dikembangkan.

1. **Simbol Flowchart**

Flowchart adalah penggambaran algoritma secara diagram yang bertujuan untuk menyelesaikan masalah dengan menggunakan sekumpulan simbol. Simbol – simbol flowchart dapat dibagi menjadi 3 kelompok, yaitu flow direction symbols, processing symbols, dan input/output symbols. Setiap simbol memiliki fungsinya masing-masing. Berikut uraian dan penjelasan simbol – simbol flowchart:

* **Flow direction symbols**

Simbol – simbol yang termasuk dalam kelompok flow direction, merupakan simbol yang digunakan untuk menghubungkan simbol satu dengan yang lain. Simbol ini disebut juga dengan connecting line. Berikut adalah simbol – simbol flow direction beserta fungsinya:

|  |  |
| --- | --- |
| **Simbol** | **Fungsi** |
| Simbol Arus | Menyatakan jalannya arus suatu proses |
| Simbol komunikasi link | Menyatakan transmisi data dari satu lokasi ke lokasi lain |
| Simbol connector | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama |
| Simbol offline connector | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda |

* **Processing symbols**

Simbol – simbol yang termasuk dalam kelompok processing symbols, merupakan simbol yang digunakan untuk menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur. Berikut adalah simbol – simbol processing symbols beserta fungsinya:

|  |  |
| --- | --- |
| **Simbol** | **Fungsi** |
| Simbol Proses | Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer |
| Simbol manual | Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer |
| Simbol decision | Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya / tidak |
| Simbol predefined process | Menyatakan suatu fungsi atau sub program |
| Simbol preparation | Menyatakan nilai awal variable atau digunakan untuk algoritma perulangan |
| Simbol terminal | Menyatakan permulaan (awal) atau akhir suatu program |

* **Input / Output symbols**

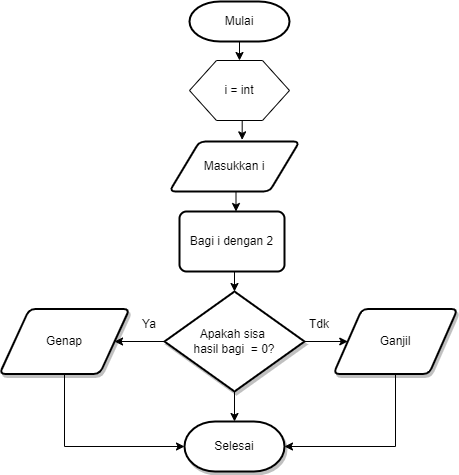
Simbol – simbol yang termasuk dalam kelompok input / output symbols, merupakan simbol yang digunakan untuk menunjukkan jenis peralatan yang digunakan sebagai media input dan output. Berikut adalah simbol – simbol input / output symbols beserta fungsinya:

|  |  |
| --- | --- |
| **Simbol** | **Fungsi** |
| Simbol input/output | Menyatakan proses input atau output tanpa tergantung jenis peralatannya |
| Simbol punched card | Menyatakan input berasal dari kartu atau output ditulis ke kartu |
| Simbol disk storage | Menyatakan input berasal dari dari disk atau output disimpan ke disk |
| Simbol document | Mencetak keluaran dalam bentuk dokumen (melalui printer) |
| Simbol display | Mencetak keluaran dalam layar monitor |

|  |  |
| --- | --- |
| **Simbol** | **Fungsi** |
| Simbol magnetic tape | Menyatakan output yang menggunakan pita magnetic |
| Simbol punched tape | Menyatakan input dan output menggunakan kertas berlubang |
| Simbol hardisk storage | Menyatakan input output menggunakan hard disk |
| Simbol magnetic drum | Menyatakan input output menggunakan drum magnetic |
| Simbol offline storage | Menyatakan file non komputer yang diarsip urut angka |
| N |
| Simbol offline storage  A | Menyatakan file non komputer yang diarsip urut huruf |
| Simbol offline storage  C | Menyatakan file non komputer yang diarsip urut tanggal |

1. **Contoh flowchart:**

Setelah mengetahui berbagai simbol yang dapat digunakan untuk membuat flowchart, maka selanjutnya dibagian ini akan diberikan contoh penggunaan simbol – simbol tersebut untuk membuat flowchart sederhana.



Flowchart diatas merupakan flowchart yang digunakan untuk menentukan apakah bilangan yang dimasukkan merupakan bilangan ganjil atau genap. Flowchart diatas diawali dengan memasukkan suatu bilangan kemudian bilangan dibagi dengan 2, selanjutnya dilakukan proses pengecekkan apakah bilangan tersebut ganjil atau genap dengan menggunakan simbol decision. Apabila bilangan tersebut dibagi dengan 2 hasilnya 0 atau habis tidak tersisa, maka bilangan tersebut termasuk bilangan genap, dan sebaliknya apabila masih ada sisanya maka bilangan tersebut merupakan bilangan ganjil.

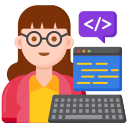
* **LATIHAN 1**

**Kerjakan soal-soal berikut dengan baik dan benar!**

1. Jelaskan pengertian Algoritma !
2. Jelaskan pengertian kalimat deskriptif, pseudocode dan flowchart!
3. Notasi kalimat deskriptif terbagi menjadi 3 bagian, sebutkan dan jelaskan!
4. Pada umumnya pseudocode menggunakan 4 jenis notasi, yaitu? Jelaskan!
5. Sebutkan Jenis – jenis Flowchart!
6. Gambarkan Simbol Proses, decision, terminal, input/output
7. Tuliskan pseudocode untuk Instruksi (pernyataan): Tulis kalimat “hello world!”

**MATERI PERTEMUAN 2**

# Membuat Flowchart Dengan Visio



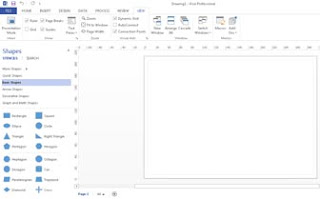
**TAHUKAH KAMU…?**

“Cara Instalasi Visio di Windows?”

“Bagaimana menggambarkan algoritma dengan Visio?”

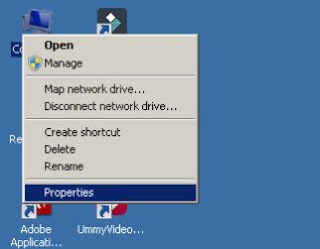
1. **Instalasi Visio di Windows**

Microsoft Office Visio adalaha salah satu varian produk Microsoft Office, yang digunakan untuk memudahkan pengguna untuk membuat diagram, baik untuk membuat atau menggambar flowchart, struktur pemetaan jaringan IT, membuat struktur organisasi atau chart organisasi, mendokumentasikan proses bisnis, atau menggambar rencana dasar, dan terkhususnya membuat Algoritma. Nah Microsoft Office Visio ini, yang memiliki fitur-fitur yang sangat lengkap untuk menyelesaikan tugas-tugas tersebut.

[](https://1.bp.blogspot.com/-uAfiAKALpq4/XmI16W1RsSI/AAAAAAAAWK8/rGOmnYiIz9MjxbTYJ_4zRc0rhUzLzJLngCLcBGAsYHQ/s1600/Cara-Instal-Microsoft-Visio-2013-32-Bit-dan-64-Bit-Berhasil.jpg)

Kali ini kita akan mencoba menginstall Microsoft Office Visio 2013 yang 32 bit, karena komputer yang kita gunakan itu 32 bit. Kalian bisa mengetahui berapa bit yang digunakan di komputer atau laptop kalian dengan cara berikut ini :

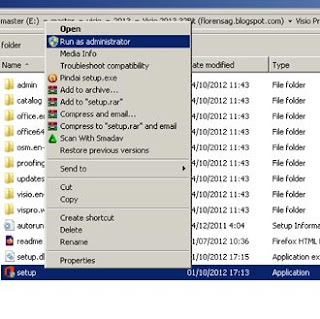
1. Klik Kanan pada My Computer.

[](https://1.bp.blogspot.com/-k3CIoDJFHFw/XmI2ORFlzqI/AAAAAAAAWLE/kwPFP83PDGcpm7IVZHmvEJEiXn741rNLACLcBGAsYHQ/s1600/cara+mengetahui+bit+yang+digunakan+di+komputer.jpg)

1. Kemudian muncul jendela System. Selanjutnya Anda dapat melihat System type, berapa bit yang digunakan di komputer atau laptop Anda. Seperti pada gambar berikut ini :

[](https://1.bp.blogspot.com/-7cvBNIolMOM/XmI2UK7DukI/AAAAAAAAWLI/P3ZkVkPWjBQrwf-XOQW7vsxH3h6JYR4qQCLcBGAsYHQ/s1600/cara+mengetahui+bit+yang+digunakan+di+komputer+atau+di+laptop.jpg)

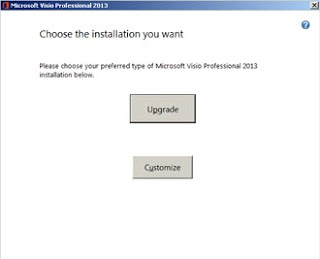
1. Klik master program Microsoft Office Visio 32 bit.

[](https://1.bp.blogspot.com/-kZCAN03sIU0/XmI2os7Y1ZI/AAAAAAAAWLU/gNiuefjtlU0IrZdocG9AmqLLBp9X-0D_gCLcBGAsYHQ/s1600/Cara-Instal-Microsoft-Visio-2013-32-Bit-dan-64-Bit.jpg)

1. Bila Anda belum mempunyai aplikasi Microsoft Office Visio 2013, silahkan Anda mendownloadnya. Selanjut muncul jendela Microsoft Visio Standar 2013, klik atau beri centang pada *I accept the terms of this agreement*, seperti gambar berikut ini :

[](https://1.bp.blogspot.com/--LbYefIugRM/XmI20BLb8cI/AAAAAAAAWLY/r5bcQ3kUuD0JFir5DrXSSbgWIQYdExHHgCLcBGAsYHQ/s1600/Cara-Instal-Microsoft-Visio-2013.jpg)

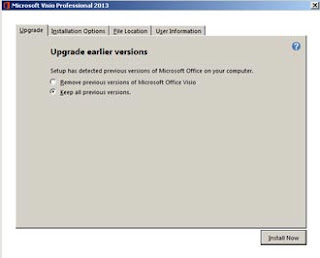
1. Pada jendela Microsoft Visio berikutnya adalah Choose the installation you want

[](https://1.bp.blogspot.com/-MuOCxygbKcg/XmI2840lr-I/AAAAAAAAWLg/Td2nx9s5XuY7JDuKSWuchcvCQUzGY0wMACLcBGAsYHQ/s1600/Cara-Instal-Microsoft-Visio.jpg)

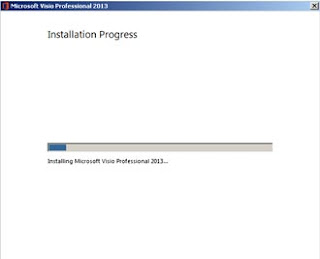
1. - Upgrade : Disii misalanya sudah ada Microsoft Visio 2010 yang sudah kita instal, dan kita ingin menggantikan dengan Microsoft Visio 2013, maka kita pilih Upgrade

- Customize : Kalau kita ingin agar program visio 2010 tetap ada dan juga program visio 2013. Nah kalau kita memilih Customize.

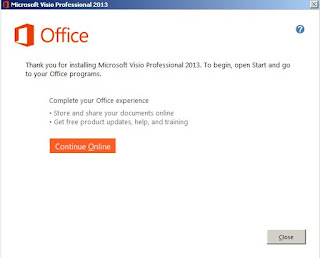
Muncul jendela Upgrade earlier version, nah kita pilih pilihan Keep all previous versions, supaya microsoft visio 2010 tetap ada. Klik Install Now.

[](https://1.bp.blogspot.com/-JQN_xZFv-fo/XmI3GpLd2hI/AAAAAAAAWLk/Qp-DT32Xx48D5T5VKJe7z_sECoKJQL0dACLcBGAsYHQ/s1600/Cara-Instal-Microsoft-Visio-2013-32-versi-Bit-dan-64-Bit.jpg)

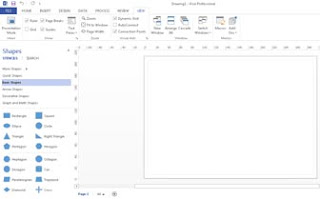
1. Tunggu beberapa saat instalnya lagi proses

[](https://1.bp.blogspot.com/-jF9Et2diIZ8/XmI3NrTGqII/AAAAAAAAWLo/kg-ON57Ey20RCbwKD7kNxik_EHRVF6yuACLcBGAsYHQ/s1600/Proses-Cara-Instal-Microsoft-Visio-2013-Berjalan.jpg)

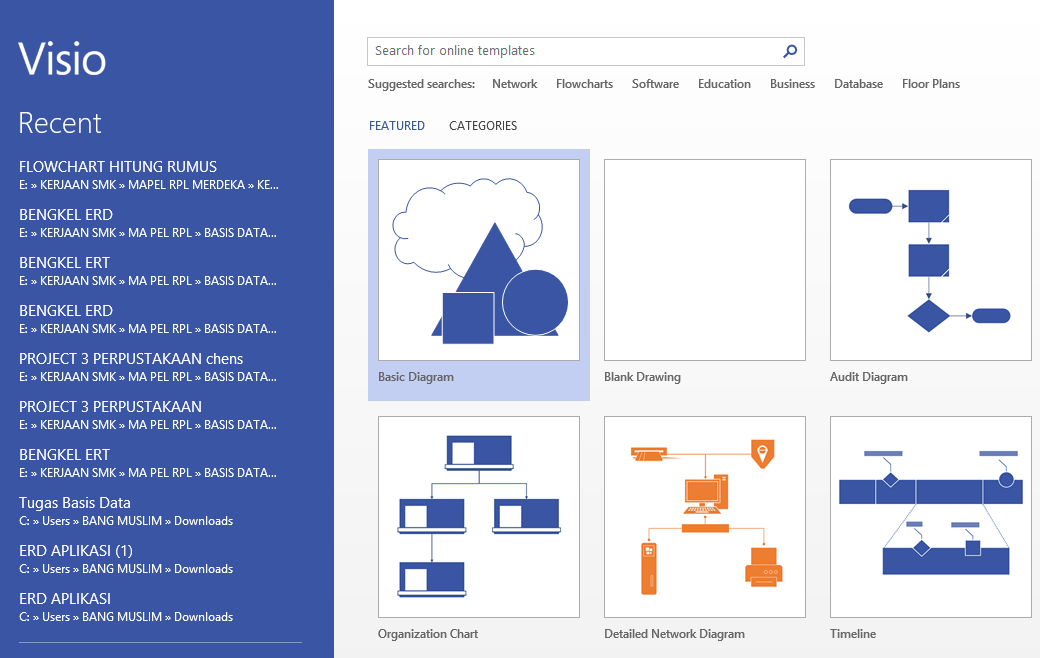
1. Terakhir klik Close

[](https://1.bp.blogspot.com/-xwN-9iZdsYY/XmI3TP6UIgI/AAAAAAAAWLw/q6yNx-etylsEucRXKfQgK-Z9VxnWEqiqACLcBGAsYHQ/s1600/Cara-Instal-Microsoft-Visio-2013-Selesai.jpg)

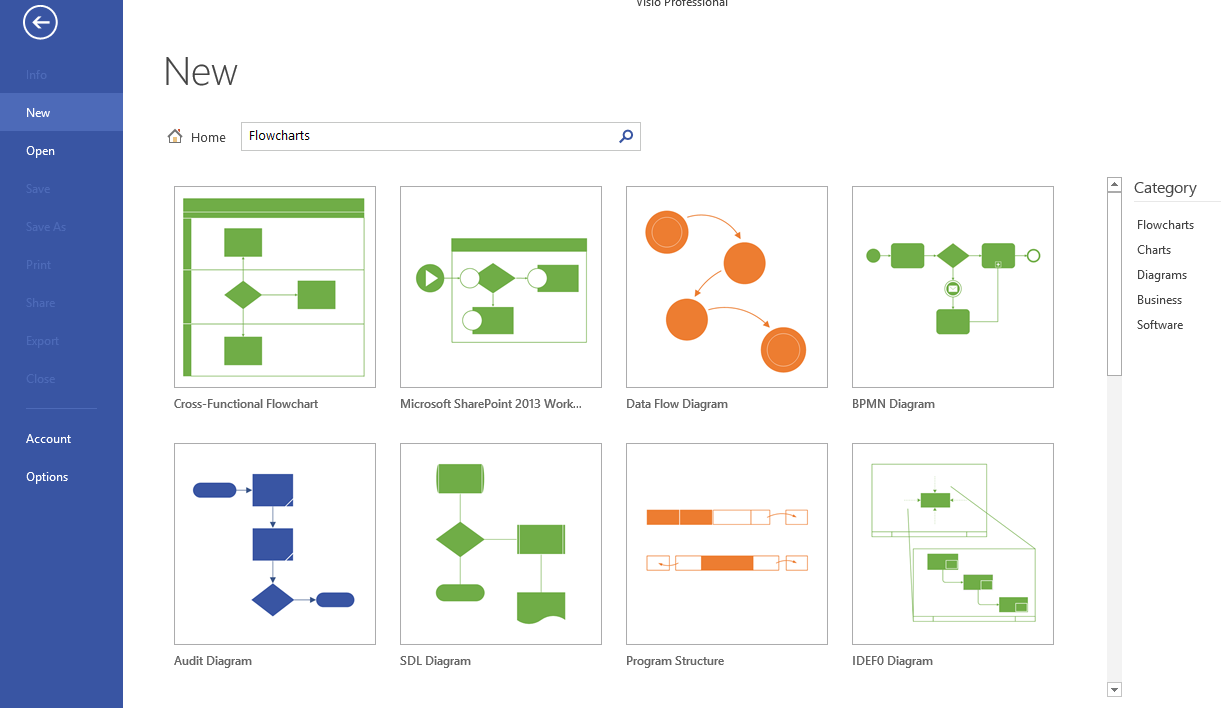
1. Sampai disini tahap cara instal visio 2013 sudah selesai, namun Microsoft Office Visio hanya dapat digunakan sesuai dengan tanggal telah ditentukan oleh programnya.

[](https://1.bp.blogspot.com/-uAfiAKALpq4/XmI16W1RsSI/AAAAAAAAWLA/un76HUhYyz4BZ2Y_uFhJfKo9QZKc-fb0QCEwYBhgL/s1600/Cara-Instal-Microsoft-Visio-2013-32-Bit-dan-64-Bit-Berhasil.jpg)

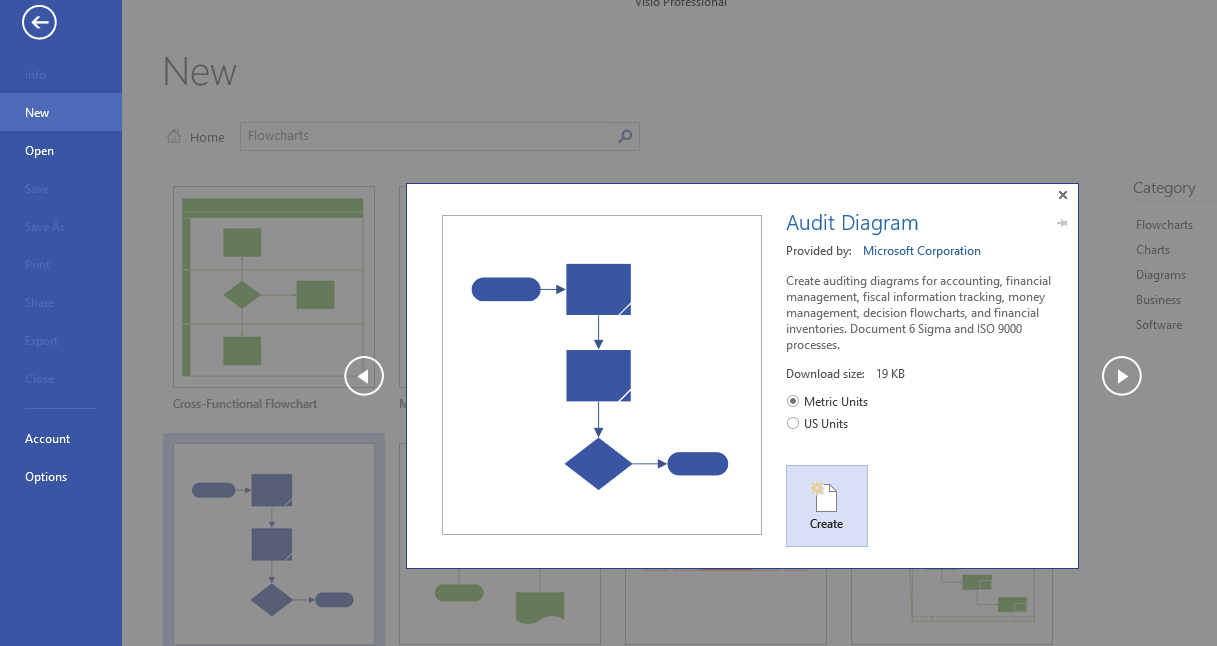
1. **Membuat Flowchart Algoritma Menggunakan Visio**
2. Buka aplikasi/program visio yang sudah diinstal sebelumnya dan akan tampil jendela sebagai berikut:



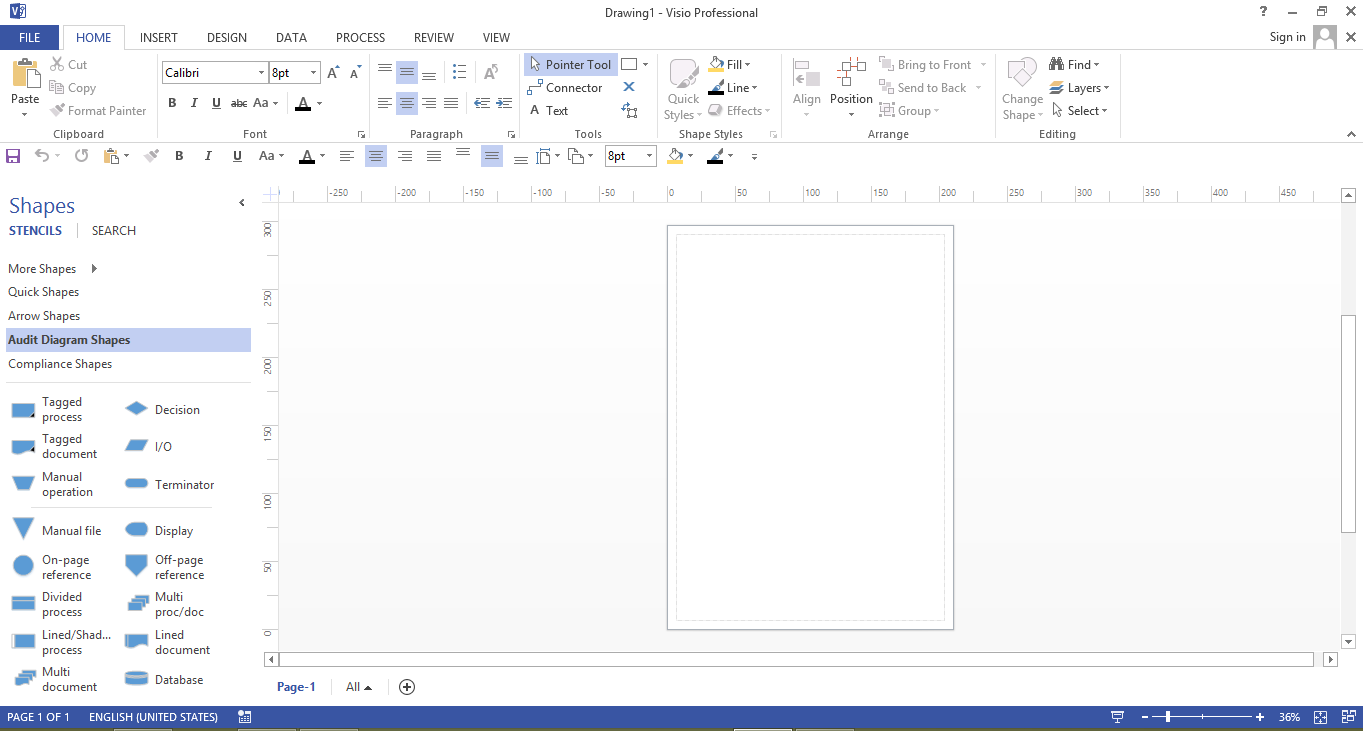
1. Pada tab suggested searched pilih flowchart, dan akan tampil jendela sebagai berikut:



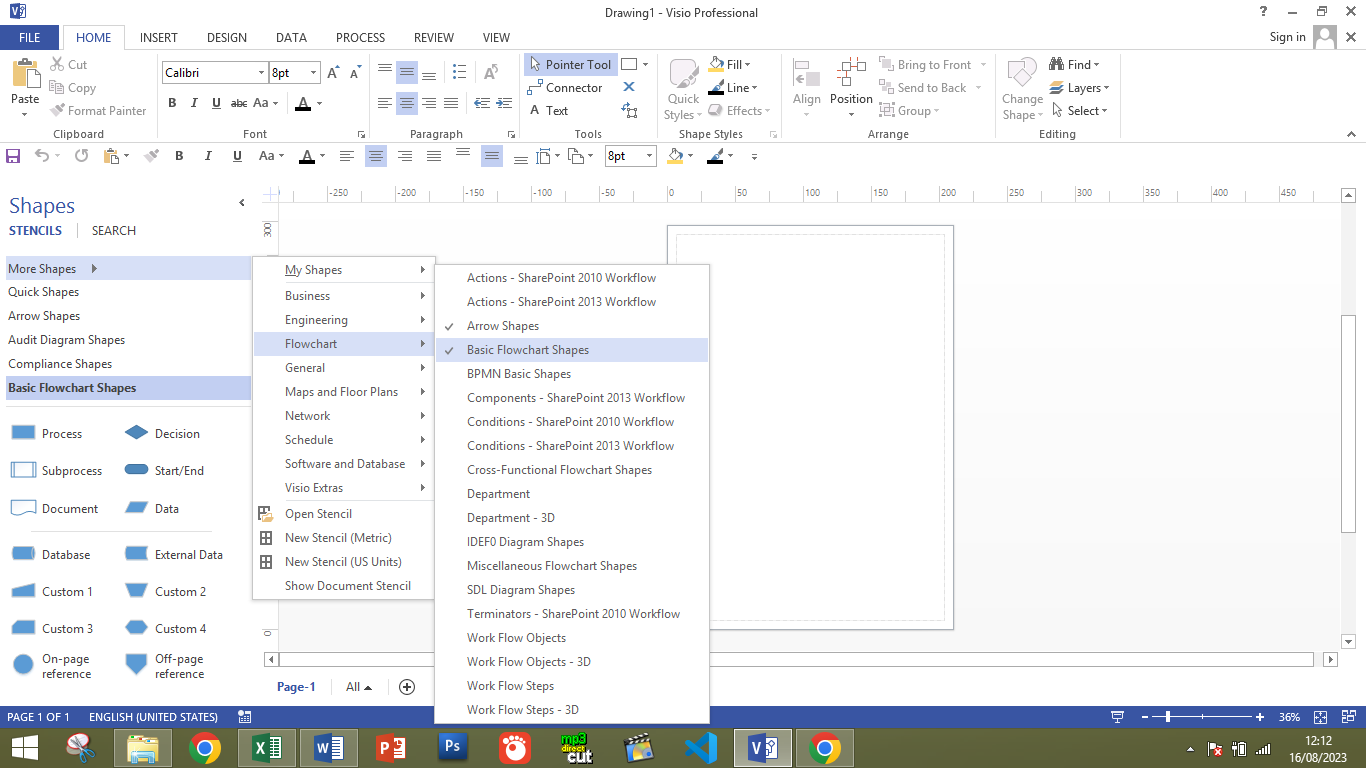
1. Pilih audit diagram, dan akan tampil jendela sebagai berikut:



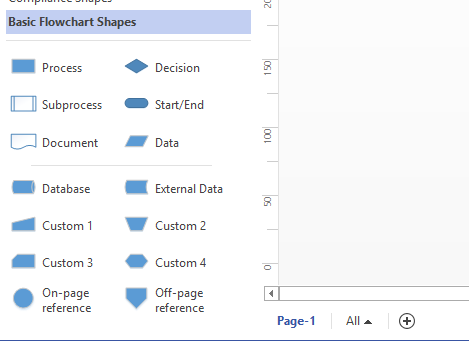
1. Biarkan ukuran dimensi pakai metric unit lalu klik create, dan akan tampil jendela visio drawing flowchart sebagai berikut:



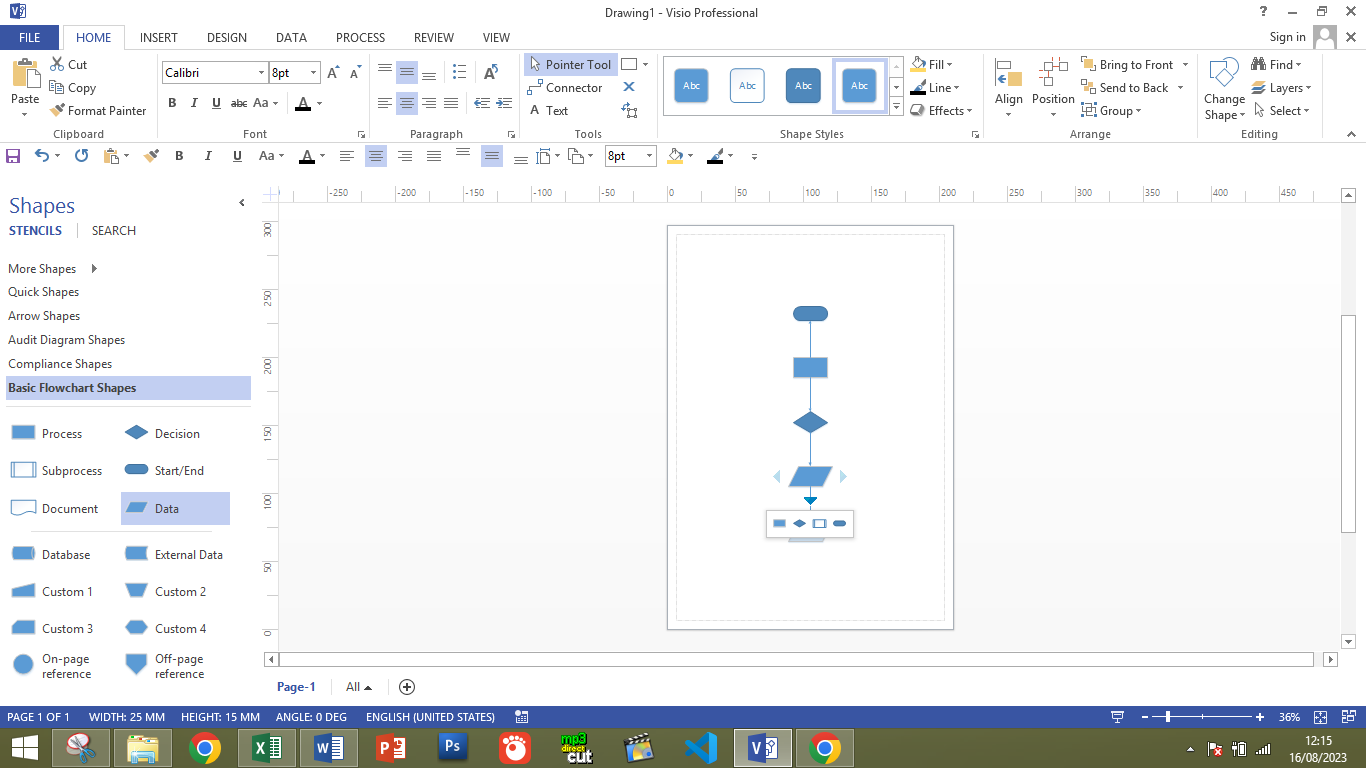
1. Untuk menambahkan shape dasar algoritma sebaiknya masukkan **basic flowchart shape** lainnya dengan cara mencentangnya di menu **more shapes**, seperti yang tampil di jendela sebagai berikut:



1. Shapes untuk menggambar notasi algoritma akan tampil seperti berikut:

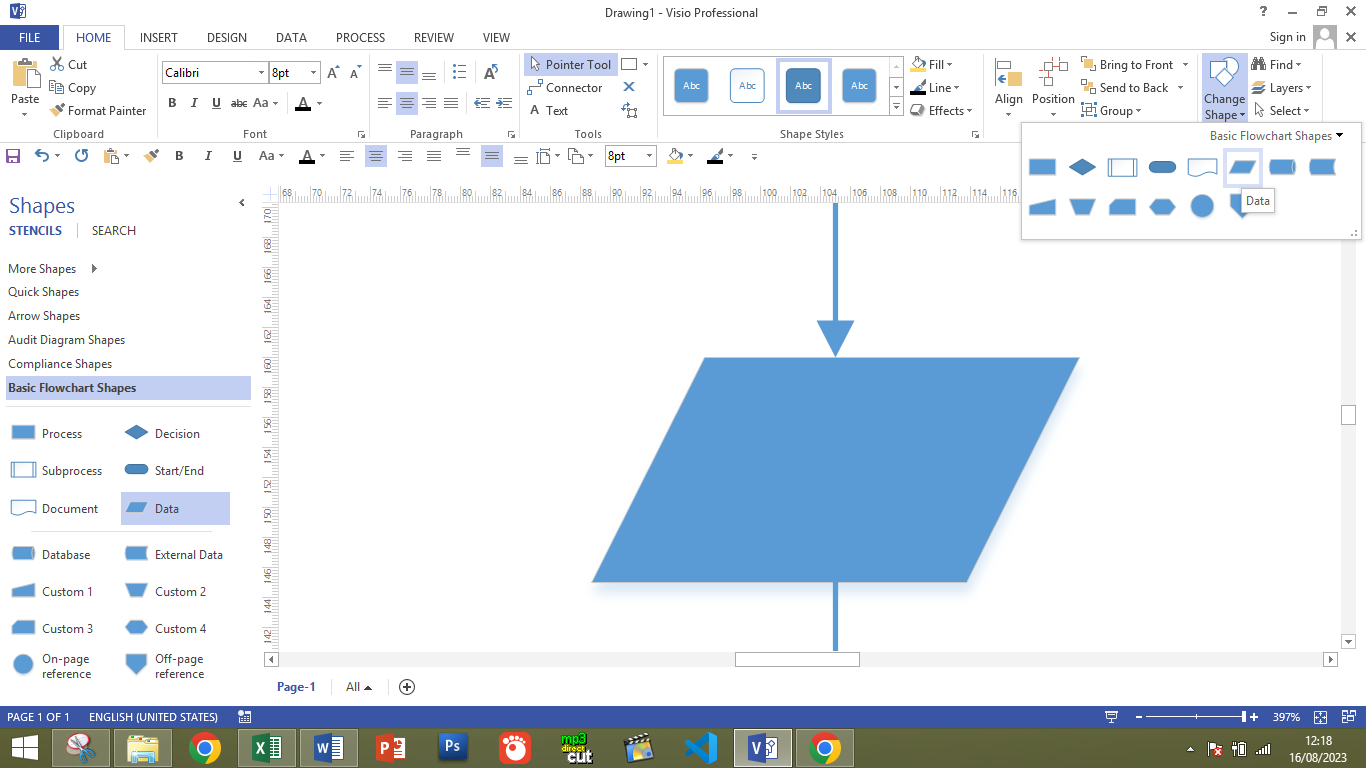


1. Mulailah menyusun shape untuk menggambarkan flowchar dengan cara Drag and Drop (seret-lepas) sebagai berikut:

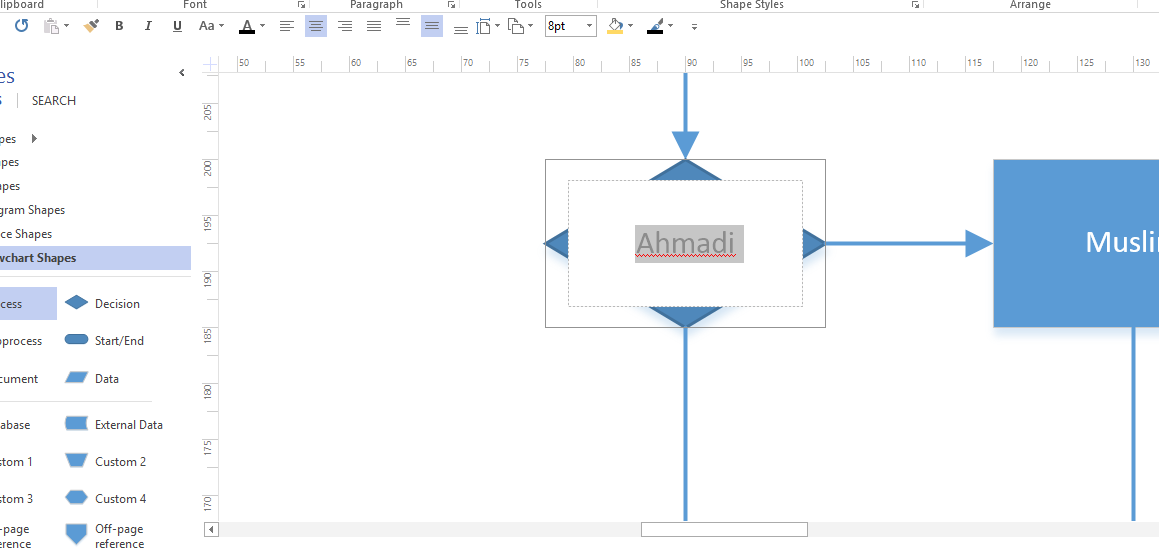


Gunakan panah semu untuk menampilkan beberapa shape yang dapat dipilih sesuai kebutuhan

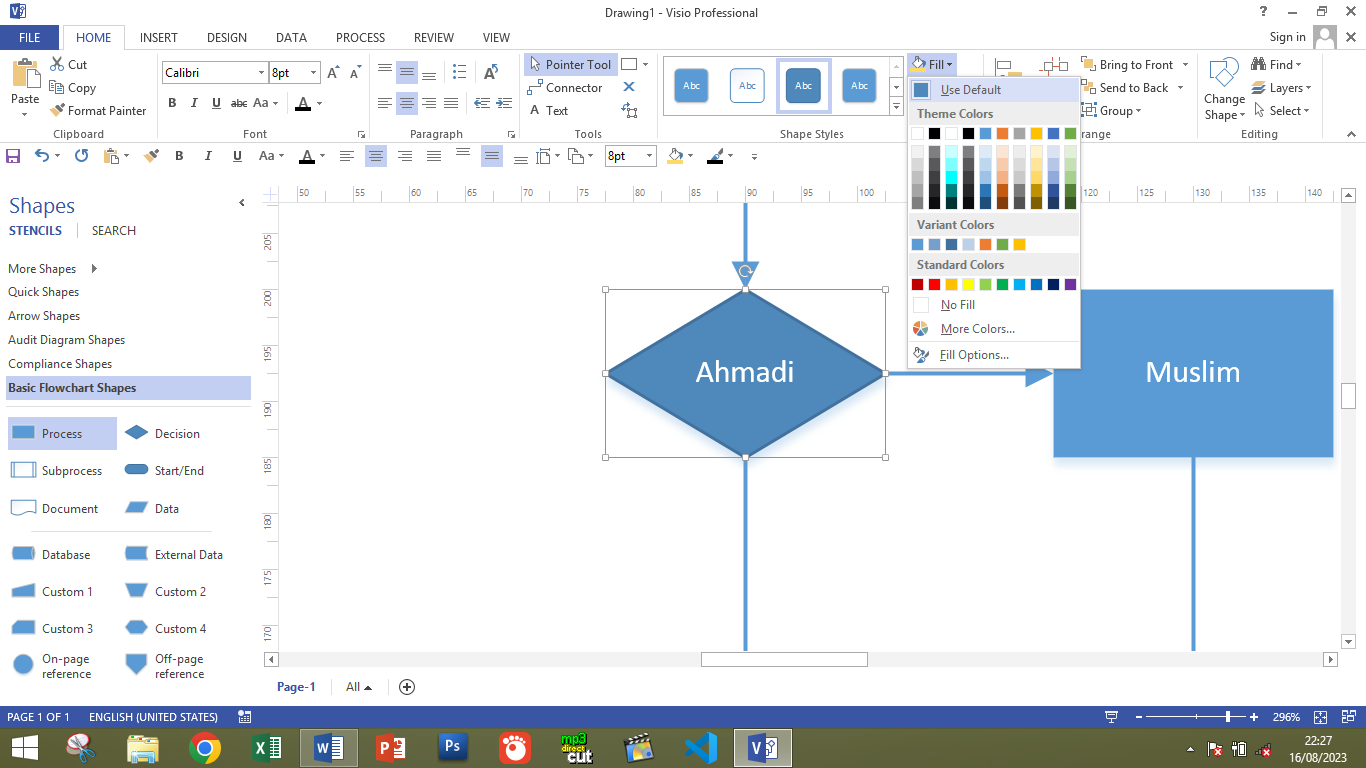
1. Jika ingin mengganti bentuk shape yang salah dan sudah terlanjur dibuat kita dapat merubahnya dengan fitur icon change shape dari menu ribbon home , seperti berikut ini:



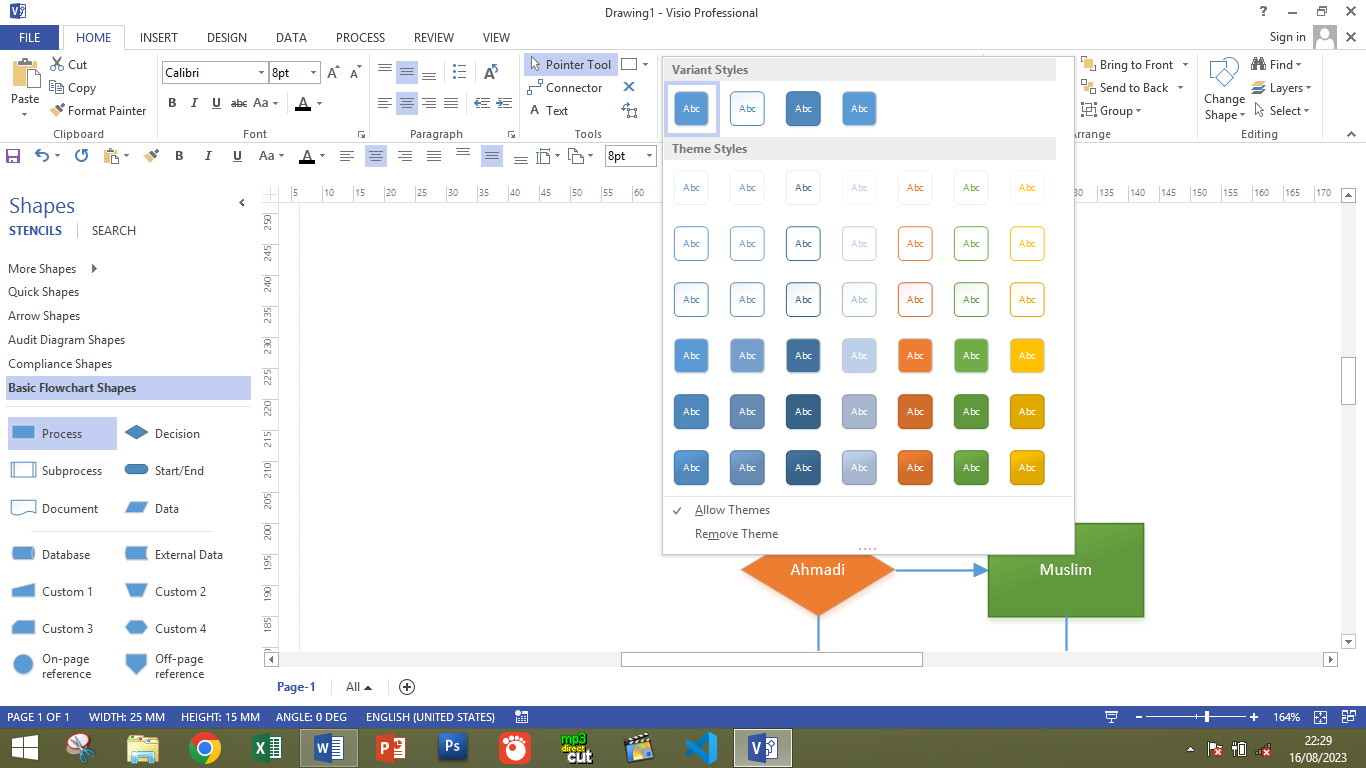
1. Jika ingin mengetik teks klik 2x pada salah satu shape , seperti berikut ini:



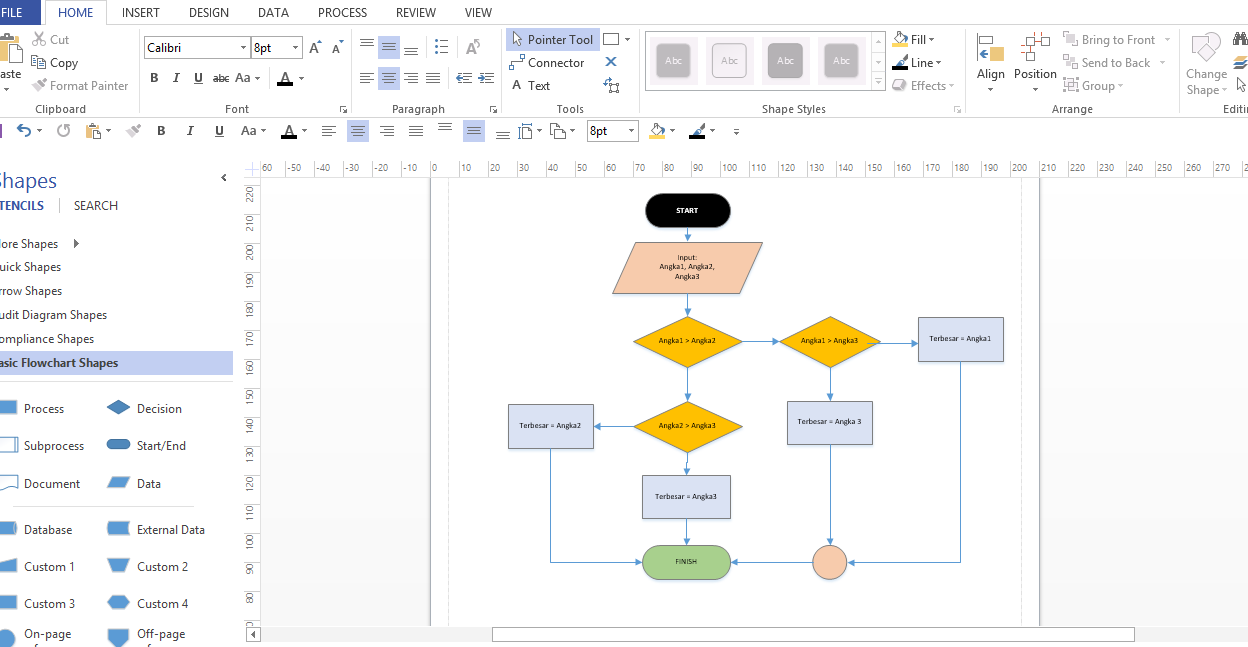
1. Jika ingin mengganti warna bisa dilakukan dengan change fill color , seperti berikut ini:



1. Atau gunakan style bawaan (template style) yang ada , seperti berikut ini:



1. Berikut ini adalah contoh flowchart algoritma yang telah berhasil dibuat:



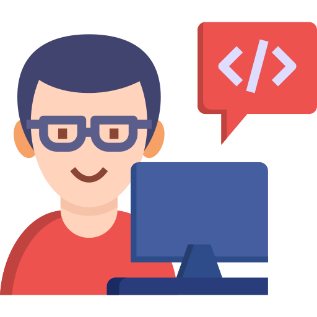
* **LATIHAN 2**

**Kerjakan soal-soal berikut dengan baik dan benar!**

1. Lakukan instalasi ms visio ke komputer
2. Jika ms office belum terinstal, maka lakukan instalasi ms office juga!
3. Buka visio yang sudah terinstal dan buatlah project algoritma sederhana!

**MATERI PERTEMUAN 3**

# Implementasi Notasi Algoritma pada Beragam Kasus



**TAHUKAH KAMU…?**

“Bagaimana Algoritma Perurutan (Squences) ?”

“Bagaimana Algoritma Percabangan (Decision) ?”

“Bagaimana Algoritma Perulangan (Iteration) ?”

1. **Algoritma perhitungan total pembayaran**

Membuat algoritma perhitungan total pembayaran pembelian sepatu merk Nevada. Dengan kondisi terdapat diskon sepatu Nevada sebesar 50%, jika pelanggan membeli sepatu merk tersebut dengan total pembelanjaan lebih dari sama dengan Rp. 500.000,-. Harga mula sepatu yang akan dibeli oleh pelanggan adalah Rp.256.000,-, dan pelanggan membeli sepatu merk Nevada sebanyak 2 pasang, sehingga untuk menghitung total pembayaran yang harus dibayar pelanggan dapat menggunakan algoritma sebagai berikut:

* **Notasi kalimat deskriptif:**

Algoritma\_hitungTotalPembayaran

{menghitung total pembayaran pembelian sepatu merk Nevada dengan memasukkan harga mula produk dan besaran diskon}

Deklarasi:

hargaMula, besaranDiskon(%), hargaDiskon = int

Deskripsi:

Input namaProduk, hargaMula, besaranDiskon(%)

Input jumlahproduk 50%,

Selain itu tidak

Hitung pembayaran = hargaMula produk \* jumlahproduk

Jika pembayaran >= 500.000 maka diberikan discount mendapat discount

Hitung total pembayaran = pembayaran – discount

Cetak total pembayaran.

* **Notasi pseudocode:**

PROGRAM PerhitunganTotalPembayaran

DEKLARASI

k = string {nama produk}

m = int {harga mula produk} n = int {jumlah produk}

d = int {besaran diskon}

p = int {hitung pembayaran}

s = int {hitung harga discount}

H = int {hitung total pembayaran}

ALGORITMA

read (k){masukkan kode produk}

read (m}{masukkan harga mula produk}

read (n){masukkan jumlah produk}

read(d){masukkan besaran discount (%)}

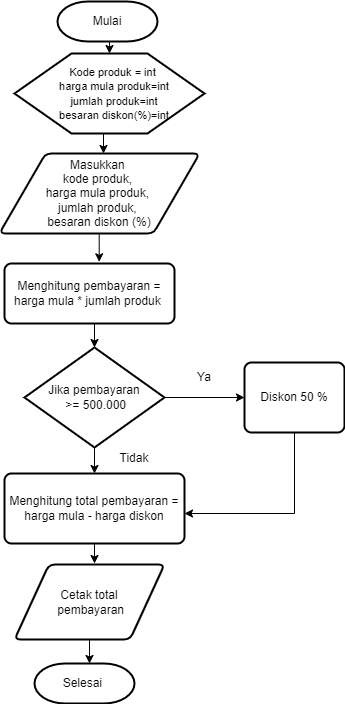
p ←m\*n

Jika p >= 500.000

maka H ← p-(p\*d)

write(H) {tampilkan total pembayaran}

End

* **Notasi flowchart:**

1. **Algoritma mengcek ketersediaan barang alat tulis di gudang**

Algoritma cek ketersediaan map bening yang ada ditoko. Dengan kondisi jika jumlah map bening kurang dari sama dengan 40 buah, maka akan dilakukan pembelian kepada supplier. Berdasarkan studi kasus, maka dapat diselesaikan dengan notasi algoritma sebagai berikut:

* **Notasi kalimat deskriptif:**

Algoritma\_HitungKetersediaanBarang

{menghitung jumlah ketersediaan barang}

Deklarasi:

Nama barang = string,

jumlah barang = int,

catatan pembelian = string

Deskripsi:

Input Nama barang

Melakukan cek ketersediaan barang Hitung jumlah barang

Jika jumlah barang <= 40 buah, maka dimasukkan dalam catatan pembelian Cetak catatan pembelian

Catatan pembelian siap digunakan acuan untuk membeli ke supplier.

* **Notasi pseudocode**

PROGRAM CekKetersediaanStok DEKLARASI

m=string (nama barang}

n=int {jumlah barang}

c=string {catatan}

ALGORITMA

read (m) {masukkan nama barang}

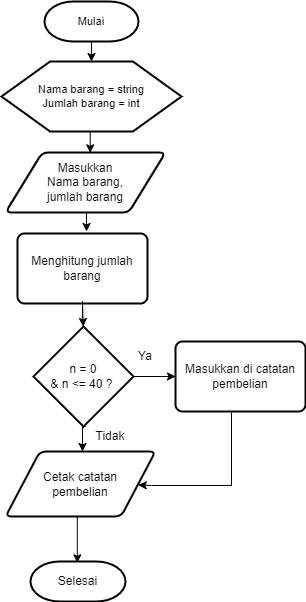
read (n} {masukkan jumlah barang}

jika n = 0 dan n <= 40

Maka c = m

write (c) {cetak Catatan pembelian}

End

* **Notasi flowchart**

1. **Algoritma menghitung gaji karyawan**

Algoritma menghitung gaji yang akan diberikan kepada karyawan, dimana gaji yang akan diterima mencakup gaji pokok, tunjangan, dan transportasi. Kasus tersebut dapat diselesaikan dengan algoritma sebagai berikut:

* **Notasi kalimat deskriptif:**

Algoritma\_hitungGaji

{menghitung gaji yang akan diberikan kepada karyawan} Deklarasi:

namakaryawan = string,

gajipokok = int,

tunjangan = int,

transportasi = int,

gajikaryawan = int

Deskripsi:

Input namakaryawan

Input gajipokok

Input tunjangan

Input transportasi

gajikaryawan = gajipokok+tunjangan+transportasi

Cetak namakaryawan diikuti dengan besar gajikaryawan

* **Notasi pseudocode**

PROGRAM gajikaryawan DEKLARASI

a=string (nama karyawan}

b=int {gaji pokok}

c=int {tunjangan}

d=int {transportasi}

e=int {gaji karyawan

ALGORITMA

read (a) {masukkan nama karyawan}

read (b} {masukkan gaji pokok}

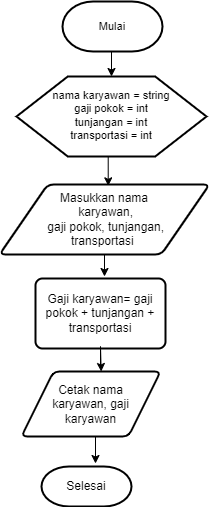
read (c} {masukkan tunjangan}

read (d} {masukkan transportasi}

e ← b + c + d

write (a,e) {cetak namakaryawan, gajikaryawan}

End

* **Notasi flowchart**

**PERURUTAN (SQUENCES)**

**KASUS: HITUNG KECEPATAN & PERCEPATAN**

* **Deskripsi Bahasa Terstruktur**

Menentukan nilai terbesar

1. Mulai

2. Masukkan angka1, angka2, dan angka3

3. Asumsikan terbesar adalah angka1 untuk awal.

4. Jika angka1 lebih besar dari terbesar maka terbesar adalah angka1

5. Jika tidak, apakah angka2 lebih besar dari terbesar, Jika ya maka terbesar sama dengan angka2;

6. Jika tidak maka terbesar sama dengan angka3.

8. Selesai

* **Flowchart**



* **Penerapan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Deskripsi** | **Pseudocode** | **English** | **Bahasa Program**  **(PHP)** |
| Mulai  Deklarasi, jarak, waktu, kecepatan, percepatan  Tentukan nilai jarak  Tentukan nilai waktu  Bagikan nilai jarak dengan waktu  Tampilkan hasil pembagian  Bagikan nilai kecepatan dengan waktu  Tampilkan hasil pembagian | **Mulai**  **Deklarasi** variabel: v, a, s, t  **masukkan** nilai: s, t  v <- s/t  a <- v/t  **Tampilkan** nilai kecepatan: v  **Tampilkan** nilai percepatan: a  **Selesai** | **start**  **declare** variabel: v, a, s, t  **input** nilai: s, t  v <- s/t  a <- v/t  **print** nilai kecepatan: v  **print** nilai percepatan: a  **finish** | <?php  $s=100;  $t=1,5;  $v=$s/$t;  echo $v;  $a=$v/$t;  echo $a; |

**PERURUTAN (SQUENCES)**

**KASUS: HITUNG LUAS PERSEGI PANJANG**

* **Deskripsi Bahasa Terstruktur**

Langkah-Langkah Menghitung Luas Persegi Panjang:

* Tentukan nilai Panjang.
* Tentukan nilai Lebar.
* Kalikan nilai Panjang dan Nilai Lebar
* Tampilkan hasil Perkalian.
* **Flowchart**



* **Penerapan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Deskripsi** | **Pseudocode** | **English** | **Bahasa program (PHP)** |
| Mulai  Deklarasi, jarak, waktu, kecepatan, percepatan  Tentukan nilai jarak  Tentukan nilai waktu  Bagikan nilai jarak dengan waktu  Tampilkan hasil pembagian  Bagikan nilai kecepatan dengan waktu  Tampilkan hasil pembagian | Mulai  **Program** **HitungLuas**  Deklarasi:  **panjang**,**lebar**,**luas**;  mulai;  **panjang**=8;  **lebar**=5;  **luas**=**panjang**\***lebar**;  cetak (**luas**);  selesai; | Start  **Program** **HitungLuas**  Declare:  **panjang**,**lebar**,**luas**;  start;  **panjang**=8;  **lebar**=5;  **luas**=**panjang**\***lebar**;  print (**luas**);  stop; | <?php  $panjang=8;  $lebar=5;  $luas=$panjang\*$lebar;  echo $luas; |

**PERCABANGAN (DECISION)**

* **Deskripsi Bahasa Terstruktur**

Menentukan nilai terbesar

1. Mulai

2. Masukkan angka1, angka2, dan angka3

3. Asumsikan terbesar adalah angka1 untuk awal.

4. Jika angka1 lebih besar dari terbesar maka terbesar adalah angka1

5. Jika tidak, apakah angka2 lebih besar dari terbesar, Jika ya maka terbesar sama dengan angka2;

6. Jika tidak maka terbesar sama dengan angka3.

8. Selesai

* **Flowchart**



* **Penerapan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Deskripsi** | **Pseudocode** | **English** | **Bahasa Program**  **(PHP)** |
| Mulai  Deklarasi, jarak, waktu, kecepatan, percepatan  Tentukan nilai jarak  Tentukan nilai waktu  Bagikan nilai jarak dengan waktu  Tampilkan hasil pembagian  Bagikan nilai kecepatan dengan waktu  Tampilkan hasil pembagian | Mulai  Masukkan **angka1**  Masukkan **angka2**  Masukkan **angka3**  terbesar <- **angka1**  Jika **angka1** > **terbesar**, maka  **terbesar** <- **angka1**  Jika angka2 > **terbesar**  **terbesar** <- **angka2**  Tapi jika tidak, maka  **terbesar** <- **angka3**  Angka **terbesar ditemukan adalah terbesar**  Selesai | Start  Input **angka1**  Input **angka2**  Input **angka3**  **terbesar** <- **angka1**  if **angka1** > **terbesar** then  terbesar <- **angka1**  else if **angka2** > **terbesar** then  **terbesar** <- **angka2**  Else  terbesar <- **angka3**  Print “**Angka terbesar :” +Terbesar**  Finish | <?php  $angka1=8;  $angka2=5;  $angka3=5;  $terbesar=$angka1;  if ($angka1>$terbesar){  $terbesar=$angka1;  } else if ($angka2>$terbesar){  $terbesar=$angka2;  else{  $terbesar=$angka3;  }  echo “bilangan terbesar adalah =”.$terbesar; |

**PERULANGAN (ITERATION)**

* **Deskripsi Bahasa Terstruktur**
  + - 1. Mulai
      2. Inisialisasi nilai awal i=0
      3. Cek Kondisi apakah nilai i <=100 jika kondisi bernilai benar maka cetak kalimat “Belajar perulangan for”
      4. Nilai i dilakukan increment (ditambahkan 1)
      5. Cek kondisi kembali apakah nilai i <=10 bila kondisi bernilai benar (true) maka cetak kembali kelimat “belajar perulangan for”
      6. Iterasi terus dilakukan hingga konisi bernilai salah (false)
      7. Selesai
* **Flowchart**

****

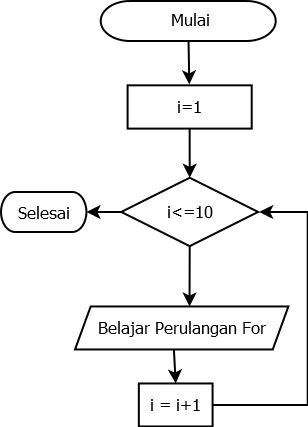
* **Penerapan**

|  |  |  |  |
| --- | --- | --- | --- |
| **DESKRIPSI** | **PSEUDOCODE** | **ENGLISH** | **BAHASA PROGRAM**  **(PHP)** |
| 1. Mulai 2. Inisialisasi nilai awal i=0 3. Cek Kondisi apakah nilai i <=100 jika kondisi bernilai benar maka cetak kalimat “Belajar perulangan for” 4. Nilai i dilakukan increment (ditambahkan 1) 5. Cek kondisi kembali apakah nilai i <=10 bila kondisi bernilai benar (true) maka cetak kembali kelimat “belajar perulangan for” 6. Iterasi terus dilakukan hingga konisi bernilai salah (false)   Selesai | **Mulai**  **Deklarasi variabel dan inisiasi** : a=0, b=100  a <- a + 1  if a > b  **Cetak** a  ke langkah 3  else //lainnya  **Selesai** | **START**  **DECLARE variabel dan inisiasi** : a=0, b=100  a <- a + 1  if a > b  **PRINT** a  ke langkah 3  else //lainnya  **FINISH**  **START**  a=0, b=100  a <- a + 1  if a > b  **PRINT** a  **Selesai** | <?php  $a = 0;  $b = 20;  $a = $a + 1;  if ($a > $b) {  echo $a . "<br>";  }  echo "Perulangan/Looping Angka 1-" . $b . "<br>";  echo "Adalah sebagai berikut : <br>";  for ($a = 0; $a <= $b; $a++) {  echo $a . "<br>";  } |

## Contoh Flowchart Perulangan di PHP

Didalam pemrograman PHP terdapat empat perulangan yang pertama perulangan for, while dan Do While. Ada juga perulangan foreach namun saya akan membahas tiga perulangan saja.

#### Flowchart Perulangan For



#### Algoritma dari flowchart diatas

1. Mulai
2. Inisialisasi nilai awal i=1
3. Cek Kondisi apakah nilai i <=10 jika kondisi bernilai benar maka cetak kalimat “Belajar perulangan for”
4. Nilai i dilakukan increment (ditambahkan 1)
5. Cek kondisi kembali apakah nilai i <=10 bila kondisi bernilai benar (true) maka cetak kembali kelimat “belajar perulangan for”
6. Iterasi terus dilakukan hingga konisi bernilai salah (false)
7. Selesai

#### Impelemtasi kedalam program PHP

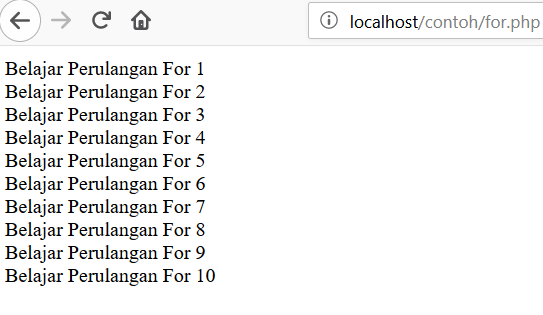
for ($i=1;$i<=10;$i++)

{

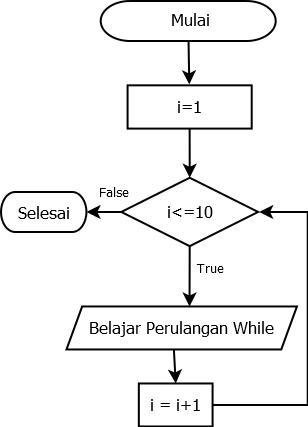
echo "Belajar Perulangan For $i <br>";

}

**Hasil Ouput:**



#### Flowchart Perulangan While



#### Algoritma dari Flowchart diatas

1. Mulai
2. Inisialisasi nilai awal i =1
3. Cek kondisi apakah i <= 10 jika ya maka cetak output “belajar perulangan while”
4. nilai i ditambahkan 1
5. cek kondisi kembali apakah i <=10 jika benar maka kembali mencetak dan nilai i ditambahkan 1
6. iterasi terus dilakukan hingga kondisi bernilai salah (false)

#### Impelentasi kedalam program PHP

$i=1;

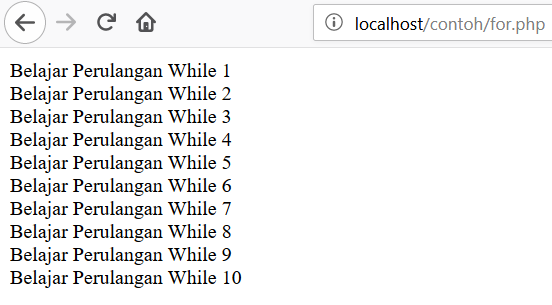
while ($i<=10)

{

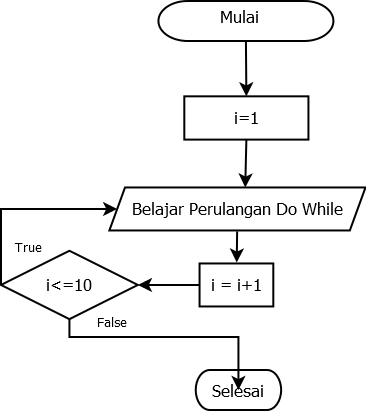
echo "Belajar Perulangan While $i <br>";

$i++;

}



#### Flowchart Perulangan Do While



#### Algoritma dari Flowchart diatas

1. Mulai
2. Inisialisasi nilai awal i =1
3. Cetak output “Belajar”
4. nilai i ditambahkan 1
5. Cek kondisi apakah nilai i <= 10, jika kondisi bernilai benar maka cetak output “Belajar Perulangan Do While”
6. Kemudian selanjutnya nilai i kembali ditambahkan 1
7. cek kondisi apakah nilai i masih <=10 jika ya maka akan dicetak output kembali
8. Iterasi terus dilakukan hingga kondisi bernilai false
9. selesai.

#### Impelementasi kedalam program PHP

$i=1;

do

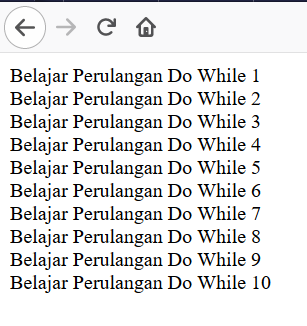
{

echo "Belajar Perulangan Do While $i <br>";

$i++;

} while ($i<=10)

Hasil Output:



* **LATIHAN 3**

**Kerjakan soal-soal berikut dengan cara mengkonversikan ke notasi Flowchart, pseudocode dan bahasa PHP!**

* Deskripsi nilai terbesar

1. Mulai

2. Masukkan angka1, angka2, dan angka3

3. Asumsikan terbesar adalah angka1 untuk awal.

4. Jika angka1 lebih besar dari terbesar maka terbesar adalah angka1

5. Jika tidak, apakah angka2 lebih besar dari terbesar, Jika ya maka terbesar sama dengan angka2;

6. Jika tidak maka terbesar sama dengan angka3.

8. Selesai

* Deskripsi hitung luas persegi panjang
  1. Tentukan nilai Panjang.
  2. Tentukan nilai Lebar.
  3. Kalikan nilai Panjang dan Nilai Lebar
  4. Tampilkan hasil Perkalian.
* Deskripsi nilai terbesar
  1. Mulai
  2. Masukkan angka1, angka2, dan angka3
  3. Asumsikan terbesar adalah angka1 untuk awal.
  4. Jika angka1 lebih besar dari terbesar maka terbesar adalah angka1
  5. Jika tidak, apakah angka2 lebih besar dari terbesar, Jika ya maka terbesar sama dengan angka2;
  6. Jika tidak maka terbesar sama dengan angka3.
  7. Selesai
* Deskripsi perulangan for

1. Mulai
2. Inisialisasi nilai awal i=0
3. Cek Kondisi apakah nilai i <=100 jika kondisi bernilai benar maka cetak kalimat “Belajar perulangan for”
4. Nilai i dilakukan increment (ditambahkan 1)
5. Cek kondisi kembali apakah nilai i <=10 bila kondisi bernilai benar (true) maka cetak kembali kelimat “belajar perulangan for”
6. Iterasi terus dilakukan hingga konisi bernilai salah (false)
7. Selesai

* Deskripsi perulangan for

1. Mulai
2. Inisialisasi nilai awal i =1
3. Cek kondisi apakah i <= 10 jika ya maka cetak output “belajar perulangan while”
4. nilai i ditambahkan 1
5. cek kondisi kembali apakah i <=10 jika benar maka kembali mencetak dan nilai i ditambahkan 1
6. iterasi terus dilakukan hingga kondisi bernilai salah (*false*)

* Deskripsi perulangan while

1. Mulai
2. Inisialisasi nilai awal i =1
3. Cetak output “Belajar”
4. nilai i ditambahkan 1
5. Cek kondisi apakah nilai i <= 10, jika kondisi bernilai benar maka cetak output “Belajar Perulangan Do While”
6. Kemudian selanjutnya nilai i kembali ditambahkan 1
7. cek kondisi apakah nilai i masih <=10 jika ya maka akan dicetak output kembali
8. Iterasi terus dilakukan hingga kondisi bernilai false
9. selesai.