MENGURUTKAN DAN MENAMPILKAN DATA YANG TERENDAH KE YANG TERTINGGI

Nama:Suryana

NIM :41513010064

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MERCU BUANA

JAKARTA

2019

Sebagai mahasiswa Universitas Mercu Buana,saya yang bertanda tangan di bawah ini:

Nama Mahasiswa :Suryana

NIM : 41513010064

Judul Tugas Akhir : Mengurutkan Dan Menampilkan Data Yang Terendah Ke Yang Tertinggi

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana Hak Bebas Royalti Noneksklusif (*None-exclusive Royalty free right)* atas karya ilmiah saya yang berjudul di atas beserta perangkat yang ada (jika diperlukan)

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan,mengalihmedia/formatkan,mengelola dalam bentuk pangkalan data(*database),* merawat dan mempublikasikan tugas akhir saya

Selain itu,demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana,saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer,Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetatp mencatmkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta

Demikian pernyataan ini saya buat dengan sebenarnya

Jakarta,10 Desember 2020

(Suryana)

LEMBAR PERSETUJUAN PENGUJI

NIM :41513010064

Nama :Suryana

Judul Tugas Akhir :Mengurutkan dan Menampilkan Data Terendah Ke data Yang Tertinggi

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika,Fakultas Ilmu Komputer,Universitas Mercu Buana

Jakarta,10-12-2020

(Sabar)

Ketua Penguji

LEMBAR PERSETUJUAN PEMBIMBING

NIM : 41513010064

Nama : Suryana

Materi Kompre ini telah diperiksa dan siap diuji sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika,Fakultas Ilmu Komputer,Universitas Mercu Buana

Menyetujui

( )

Dosen Pembimbing

Mengetahui,

(Dicky Firdaus,S Kom,MM)

Koord.Tugas Akhir Teknik Informatika

(Desi Ramayanti,S.Kom,MT)

ka.Prodi Teknik Informatika

ABSTRAK

Nama :Suryana

NIM :41513010064

Pembimbing TA:Dicky Firdaus

Judul :Mengurutkan Dan Menampilkan Data Terendah Ke Data Yang Tertinggi

Saya membuat materi ini sesuai tugas kompre yang telah diajarkan kepada saya.Tujuan membuat penelitian ini untuk membuat tugas akhir saya metode penelitian ini dibuat dengan model sdlc kesimpulan dari materi ini membuat program output pencarian dari data yang sudah tersimpan dan mengurutkannya

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat allah swt karena berkat rahmatnya saya telah menyelesaikan tugas akhir ini

Penulis menyadari bahwa tanpa bantuan dan bimbingan dosen tidak akan bisa membuat tugas akhir ini oleh karena itu,penulis mengucapkan terimakasih kepada

1.Pembimbing tugas akhir kompre saya

2.Staf-staf yang ikut membantu

Akhir kata mudah-mudahan tugas akhir saya dan dosen –dosen yang membantu dapat bermanfaat

Jakarta 10-12-2020

Aplikasi Mengururkan Data Dari Yang Terendah Ke Yang Tertinggi

Latar Belakang

Meminjam buku secara manual membuat sulit karena banyaknya orang yang sekolah dan jika meminjam buku diperlukan komputer untuk mengoperasikannya Peminjaman buku adalah yang sering dilakukan di setiap instansi sekolah

Tujuan dan manfaat

Agar semakin mudah dalam menginput buku

Batasan masalah

Hanya membahas pinjaman

Landasan Teori

Landasan teori ini ada binary search algoritma sorting

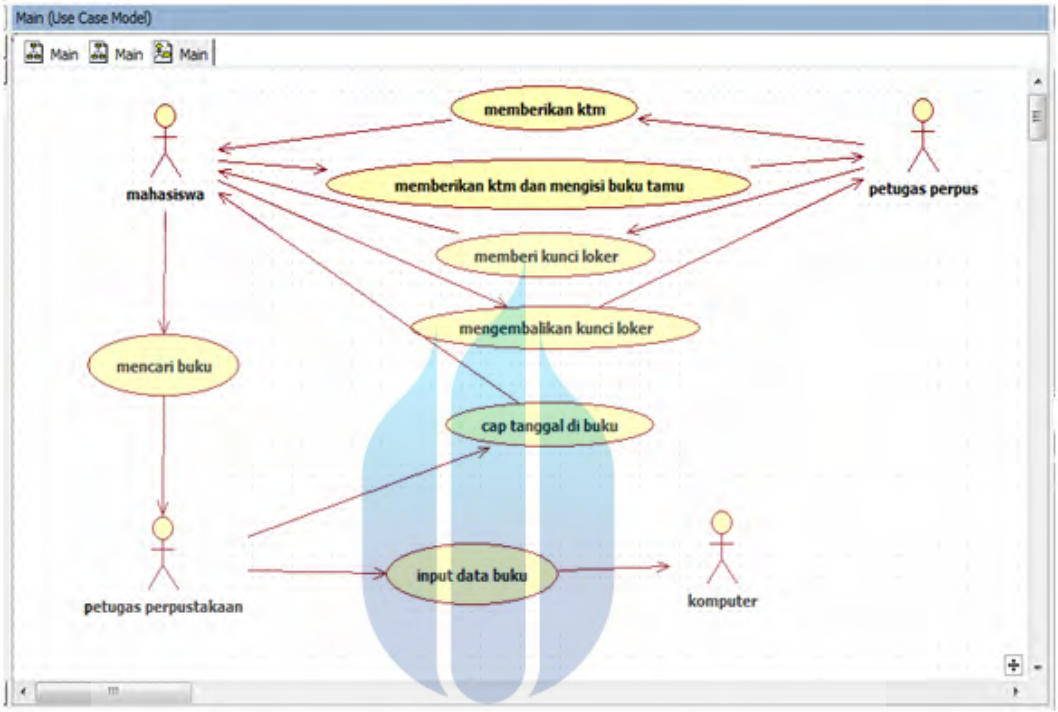
Analisa Masalah

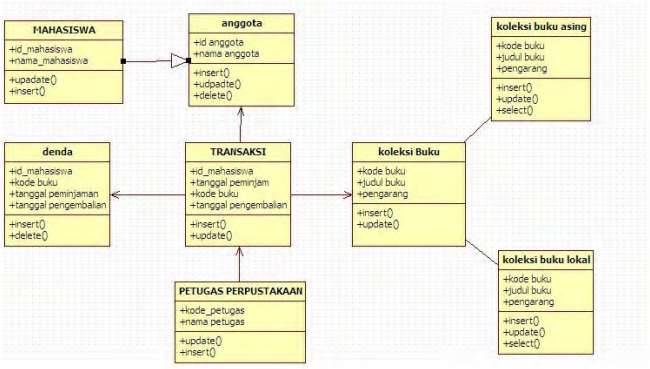
Algoritma strukur data flowchart kodingan

Perancangan sistem

Membuat topik mencari materi membuat kodingan dan menguji

Perancangan uml





Perancangan database

Data Buku

Field Name	Data Type
Kode Buku	Text
Judul Buku	Text
Pengarang	Text
Penerbit	Text
Tahun Terbit	Text

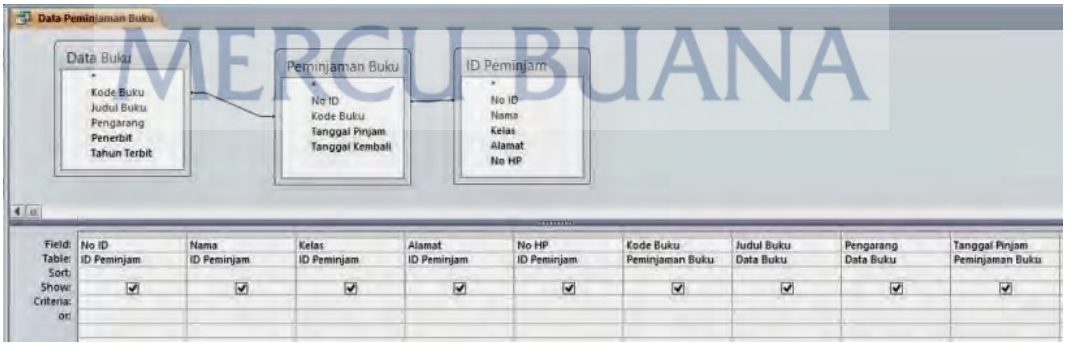
Data Buku | ID Peminjam | Peminjaman Buku

Field Name	Data Type
No ID	Text
Kode Buku	Text
Tanggal Pinjam	Date/Time
Tanggal Kembali	Date/Time

Data Buku | ID Peminjam | Peminjaman Buku

Field Name	Data Type
No ID	Text
Kode Buku	Text
Tanggal Pinjam	Date/Time
Tanggal Kembali	Date/Time

www.belajaroffice.com



Perancangan interface

Definisi Algoritma Sorting

Istilah pengurutan dalam pemrograman lebih dikenal dalam bahasa inggris atau disebut *Sorting*.Fungsi *Sorting* adalah agar data yang tersedia dapat menjadi *Sorted* atau terurut menurut kaidah/aturan tertentu Pengurutan data dalam struktur data sangat penting, baik untuk data yang bertipe data numerik maupun karakter. Pengurutan dapat dilakukan secara ascending (naik) maupun descending (turun). Pengurutan adalah proses menyusun kembali data yang acak menjadi susunan yang teratur menurut aturan tertentu. Berikut adalah metode-metode yang telah ada dalam hal *Sorting* :*Exchange Sort,Selection Sort,Insetion Sort,Bubble Sort,Quick Sort,Shell Sort,Binary Insertion Sort,* dan lain-lain

Shell Sort adalah pengurutan yang lebih baik dalam mengenai banyak data.Shell Sort menggunakan metode perbandingan dan pertukaran.Metode ini disebut juga dengan metode pertambahan menurun (diminishing increment). Metode ini dikembangkan oleh Donald L Shell pada tahun 1959, sehingga sering disebut Metode Shell Sort.Perbandingan dimulai dari separuh array yang akan disortir dengan separuh bagian yang lain. Contoh:Jika terdapat 100 elemen,diperbandingkan elemen 1 dan elemen 51,elemen 2 dan elemen 52 dst.Selanjutnya algoritma akan membandingkan elemen 1 dan elemen 26,elemen 2 dan elemen 27 dst.

Penjelasan Algoritma Shell Sort

1. Program akan dijalankan jika range < > 0 terpenuhi
2. Sebelum masuk putaran ditentukan range dan target
3. Pada putaran ke-1,range =banyak/2
4. Tiap putaran dimulai dari counter = 1 sampai dengan counter = target
5. Pada tiap counter dilakukan proses:kiri=counter dan selanjutnya,
6. Item(kiri) dibandingkan dengan item(kanan) dimana: kanan=kiri+range
7. Jika item(kiri) >= item(kanan) maka proses selesai dan dilanjutkan counter atau mungkin putaran berikutnya
8. Jika item(kiri) < item(kanan) maka terjadi pertukaran,selanjutnya:
9. Jika item kiri < range maka proses selesai dan dilanjutkan counter berikutnya

10. Jika kiri > range maka kiri = kiri - range dan proses dimulai dari awal perbandingan item(kiri) dan item(kanan) lagi
11. Jika semua counter pada suatu putaran telah selesai maka range akan dihitung kembali yaitu : range=range/2
12. Jika range < > 0 maka program akan dijalankan sampai range=0.Berarti data telah terurut

a.Membuat program sorting aray dengan metode Shell Sort

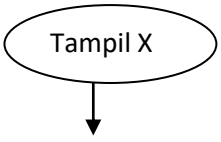
1. **Algoritma program sorting array dengan metode Shell Sort**

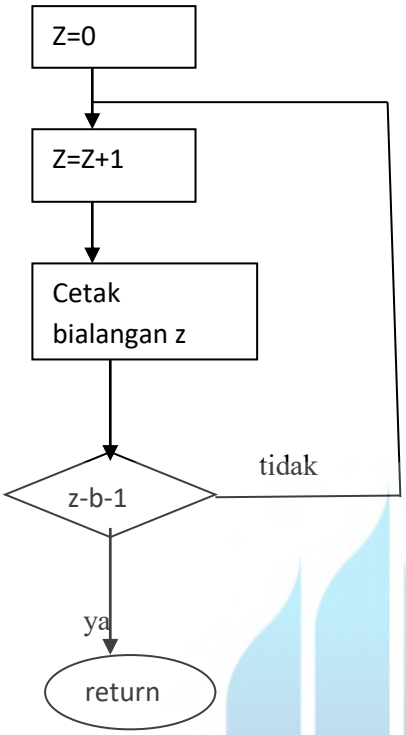
- **Algoritma Subrutin Tukar**
  1. Tukar A1 dan A2
  2. Memasukkan data pada variable A1 pada variable bantuan A3
  3. Memasukkan data paa variable A2 pada variable A1
  4. Memasukkan data pada variable bantuan A3 pada variable A2
  5. Return
- **Algoritma Subrutin Tampil**
  1. Tampil
  2. Membuat variable Z dengan based angka -1 untuk menentukkan index array yang akan ditampilkan
  3. Menambahkan Z dengan 1
  4. Menampilkan array dengan index variable Z
  5. Apakah variable Z sama dengan banyak data acak yang telah disimpan di variable B-1?
  6. Jika ya Step 7,Jika tidak Step 3
  7. Return

- **Algoritma Program Utama**
  1. Start
  2. Menciptakan variable array bilangan dari index 0 hingga 100
  3. Menciptakan variable A untuk menentukan index array yang nantinya akan dimasukan data inputan
  4. Menciptakan variable B untuk pembatas index array maksimal
  5. Membaca Inputan jumlah data total dan dimasukkan pada variable B

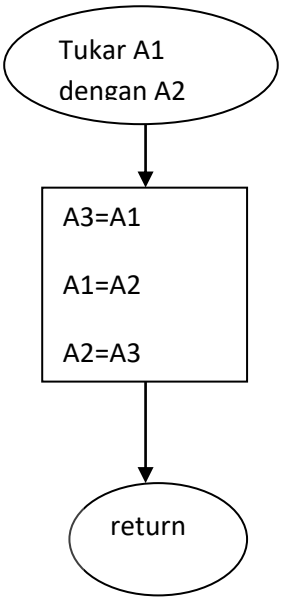
6. Menyimpan data inputan ke variable Array dengan index
7. Menambahkan Index Array/ variable A dengan 1
8. Apakah index array(A) sama dengan jumlah data total?
9. Jika ya step 10, Jika tidak step 5
10. Setting nilai awal x=0 dan nilai maksimal = jumlah data yang ada atau variable B
11. Membagi jumlah data menjadi 2 buah grup sehingga didapat nilai range atau gap(range=max\2)
12. Membandingkan data array pada indek[x] dengan data pada indek[x+range]
13. Apakah data pada index[x] lebih besar dari data yang ada pada index[x+range]?
14. Jika Ya Step 15 Jika tidak step 22
15. Melakukan pemanggilan subrutin Tukar antara bilangan[x] dengan data bilangan [x+range]
16. Dengan terjadinya penukaran maka variable adatukar yang bertipe Boolean akan bernilai true
17. Apakah adatukar adalalah "true" dan nilai range lebih kecil dari maksimal dikurangi range(range < B – range) dan nilai x lebih besar sama dengan range(x>=range)
18. Jika Ya Step 19 Jika Tidak step 22
19. Mengurangi nilai x dengan nilai range(x = x – range)
20. Mengembalikkan nilai Boolean variable adatukar menjadi false
21. Kembali ke step 12
22. Menambah nilai x dengan 1(x=x+1)
23. Membagi nilai range dengan 2 range = range\2)
24. Apakah nilai range=0?
25. Jika Ya kestep 26,jikatidak kembali ke step 12
26. Stop

Flowchart Program



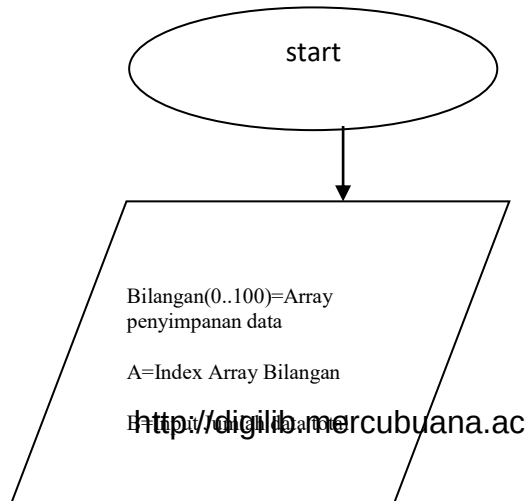


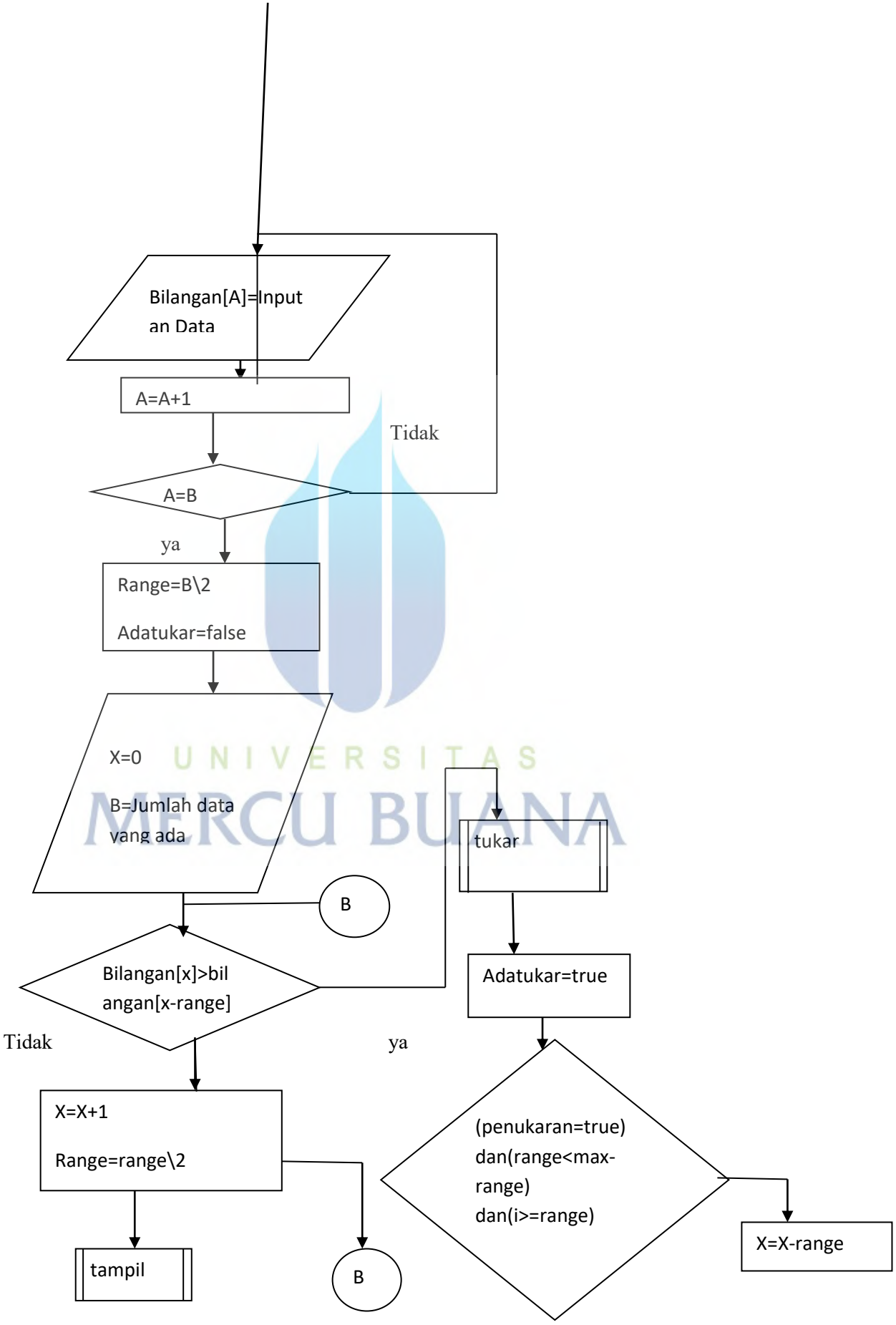
a)Sub Rutin Tukar

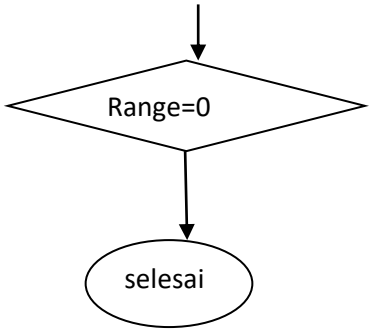


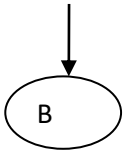
b)Sub Rutin Tampil

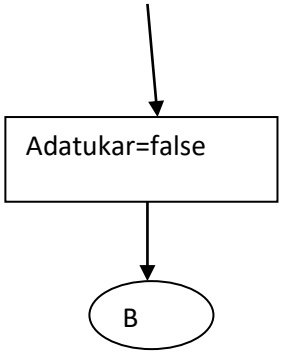
Flowchart











Source Code

```
uses wincrt;
var
   bilangan:array[0..100] of integer;
    awal,range,A,B,X,Y,Z,T:integer;
    adatukar:boolean;
    label back;
procedure tukar(var A1,A2:interger);
    var A3:integer;
     begin
     A3:=A1;
     A1:=A2;
     A2:=A3;
      end;
procedure tampil(var x:integer);
   begin
      for z:0 to  B-1 do
         write(bilangan [z],`  `);
          writeln;
           end;
 begin
    write('masukan banyak bilangan terlebih dahulu:');
read(B);
writeln('');
```

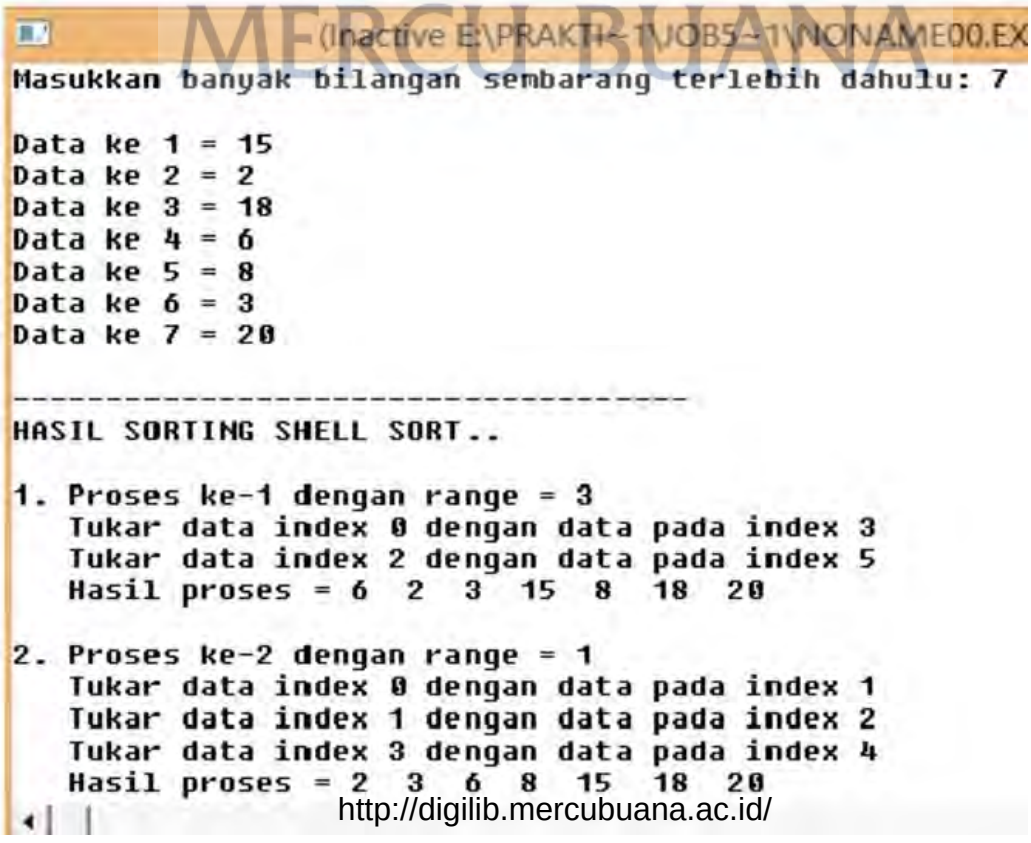
```
Writeln(‘ ‘);
Range:=B div 2;
Adatukar:=false;
t:=1;
repeat
 begin
write(t,’.proses ke-‘,t,’dengan range=’,range);
writeln;
for x:=0 to B-range-1 do
back:
begin
  if bilangan [x] >bilangan [x+range] then
 begin
   writeln (‘tukar data index’,x,’dengan data pada index’,x+range);
 tukar (bilangan[x],bilangan[x+range]);
adatukar:=true;
if (adatukar=true)and(range<B-range) and(x>=range) then
                    begin
                    adatukar:=false;
                   x:=x-range;
                  go to back;
```

```
Until range=0

end
```

Keterangan:

Pada program di atas,apabila dilakukan inputan yang merupakan variable B akan dilakukan perulangan yang dilakukan untuk mengisi input data yang dimasukkan ke dalam variable array bilangan. Setelah data acak/sembarang dimasukkan dan disimpan pada array,selanjutnya data ini ditilik dan hendak ditampilkan dengan cara urut dengan bantuan perulangan berdasarkan algoritma yang telah dibuat. Setelah seleesai diurutkan akan ditampilkan data dalam bentuk *sorted*/terurut

d. Output Program



Keterangan:

Hasil dari shell sort pada

gambar 3 diatas dimana data urut ditaruh pada hasil proses yang terakhir yaitu proses ke-2.Berikut adalah penjelasan Shell Sort setiap proses:

Proses ke-1

Nilai jumlah data awal yaitu 7

Sehingga Nilai Range =7\2=3

- **Data Awal adalah sebagai berikut:**

  Index : 0 1 2 3 4 5 6

  Data : 1 2 1 6 8 3 2

  5 8 0

- **Melakukan Perbandingan Nilai tiap index dengan tiap index+range:**

  1.Index 0 dengan Index 3

  Karena index 0 lebih besar dari index 3 maka ditukar

  Hasil : 6 2 1 1 8 3 2

  2 Index 1 dengan Index 4

  Karena index 1 lebih kecil dari index 4 maka tidak ditukar

  Hasil :6 2 1 1 8 3 2

  8 5 0

  3.Index 2 dengan Index 5

  Karena index 2 lebih besar dari index 5 maka ditukar

  Hasil : 6 2 3 1 8 1 2

  5 8 0

  4.Index 3 dengan Index 6

  Karena index 3 lebih kecil dari index 6 maka tidak ditukar

  Hasil : 6 2 3 1 8 1 2

  5 8 0

  Hasil Akhir Proses 1 adalah hasil pada poin 4 yaitu setelah index mencapai index maksimalnya,index maksimal dirumuskan adalah berasal dari jumlah index maksimal dikurangi range dan 1

2.Proses ke-2

Nilai range didapat dari nilai range sebelumnya dengan hasil dibagi 2

3\2=1

- **Data Awal adalah sebagai berikut(Berasal dari Proses-1)**

  Index : 0 1 2 3 4 5 6

  Data : 6 2 3 1 8 1 2

  5 8 0

- **Melakukan Perbandingan Nilai tiap index dengan tiap index+range:**

1.Index 0 dengan Index 1

Karena index 0 lebih besar dari index 1 maka ditukar

Hasil : 2 6 3 1 8 1 2

5 8 0

2.Index 1 dengan Index 2

Karena index 1 lebih besar dari index 2 maka ditukar

Hasil : 2 3 6 1 8 1 2

5 8 0

Karena adanya penukaran dan pada proses ini dengan range 1 maka setiap index dapat

Dibandingkan ke kanan dan ke kiri maka kembali dilakukan perbandingan dengan index

Sebelah kirinya lagi:

- **Index 1 dengan index 0**

Karena index 1 lebih kecil dari index 0 maka tidak ditukarkan

Hasil : 2 3 6 1 8 1 2

5 8 0

3.Index 2 dengan Index 3

Karena index 2 lebih kecil dari index 3 maka tidak ditukar

Hasil : 2 3 6 1 8 1 2

5 8 0

4.Index 3 dengan Index 4

Karena index 3 lebih besar dari index 4 maka ditukar

Hasil : 2 3 6 1 8 1 2

5 8 0

Karena adanya penukaran dan pada proses ini dengan range 1 maka setiap index dapat dibandingkan ke kanan dan ke kiri maka kembali dilakukan perbandingan dengan index sebelah kirinya lagi:

- **Index 2 dengan index 3**

  Karena index 2 lebih kecil dari index 3 maka tidak ditukar

  Hasil : 2 3 6 1 8 1 2

  5 8 0

- **Index 1 dengan index 2**

  Karena index 1 lebih kecil dari index 2 maka tidak ditukar

  Hasil : 2 3 6 1 8 1 2

  5 8 0

- **Index 0 dengan index 1**

  Karena index 0 lebuh kecil dari index 1 maka tidak ditukar

  Hasil : 2 3 6 1 8 1 2

  5 8 0

5. Index 4 dengan Index 5

Karena index 4 lebih kecil dari pada index 5 maka tidak ditukar

Hasil : 2 3 6 1 8 1 2

5 8 0

6. Index 5 dengan Index 6

Karena index 0 lebih kecil dari index 1 maka tidak ditukar

Hasil : 2 3 6 1 8 1 2

5 8 0

Hasil akhir proses 2 adalah hasil pada poin 6 yaitu setelah index mencapai index maksimalnya, index maksimal dirumuskan adalah berasal dari jumlah indek maksimal dikurangi range dan 1.Pada proses kedua ini nilai range telah mencapai 1 sehingga perbandingan telah selesai dan telah tercipta data baru yang telah terurut

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan tahap normalisasi dimulai dari tahap paling ringan(INF) hingga paling ketat (5NF) biasanya hanya sampai pada tingkat 3NF

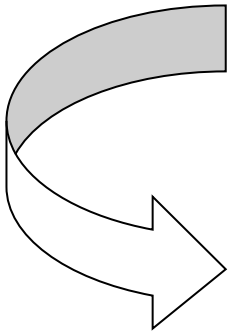
NORMALISASI

Sebuah tabel dikatakan baik(efisien) atau normal jika memenuhi 3 kriteria sbb:

1 Jika ada dekomposisi (penguraian) tabel,maka dekomposisinya harus dijamin aman*(Lossless-Join Decomposition)*.Artinya,setelah tabel tersebut diuraikan/didekomposisi menjadi tabel-tabel baru,tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis
2 Terpeliharanya ketergantungan fungsional pada saat perubahan data(Dependency Preservation)

Tabel Universal(Universal/ Star Table) ⟶ sebuah tabel yang merangkum semua kelompok data yang saling berhubungan, bukan merupakan tabel yang baik

Misalnya:



Functional Dependency

Notasi:A ⟶ B

A dan B adalah atribut dari sebuah tabel.Berarti secara fungsional A menentukan B atau B tergantung pada A,jika dan hanya jika ada 2 baris data dengan nilai A yang sama,maka nilai B juga sama

Notasi:A $\not\rightarrow$ B atau A x $\longrightarrow$ B

Adalah kebalikan dari notasi sebelumnya

Functional Dependency

Contoh tabel nilai

NamaKul	Nrp	namaMhs	NiHuruf
Struktur Data	99909	Aji	B
Bahasa Indonesia	99909	Aji	B
Basis Data	99907	Bella	A

Functional Dependency

Functional Dependency dari tabel nilai

Nrp $\longrightarrow$ namaMhs

Karena untuk setiap nilai nrp yang sama,maka nilai namaMhs juga sama

{NamaKul,nrp}$\longrightarrow$ NiHuruf

Karena atribut NiHuruf tergantung pada NamaKul dan nrp secara bersama-sama.Dalam arti lain untuk NamaKul dan nrp merupakan key(bersifat unik)

NamaKul $\not\rightarrow$ nrp

Nrp $\not\rightarrow$ NiHuruf

Bentuk-bentuk Normal

1. Bentuk Normal Tahap Pertama (1st Normal Form/1NF)
2. Bentuk Normal Tahap Kedua (2nd Normal Form/2NF)
3. Bentuk Normal Tahap (3rd Normal Form/3NF)

4. Bentuk Normal Tahap (4th Normal Form/4NF)
5. Bentuk Normal Tahap (5th Normal Form/5NF)

Bentuk Normal Tahap Pertama

(1st Normal Form /1 NF)

Bentuk normal 1NF terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*multivalued attribute),*atribut composite atau kombinasinya dalam domain data yang sama

Setiap atribut dalam tabel tersebut harus bernilai *atomic*(tidak dapat dibagi-bagi lagi)

Contoh

Misal data mahasiswa sbb:

Nrp	nama	Hobi
4445	Jami	Berenang,basket
4446	Sinta	olahraga

Atau

Nrp	nama	Hobi 1	Hobi 2
4445	Jami	berenang	basket
4446	Sinta	olahraga	

Tabel-tabel diatas tidak memenuhi syarat 1NF

Contoh

Didekomposisi menjadi

Nrp	Nama
4445	Jami

Tabel Hobi

Nrp	Hobi
4445	berenang

4445	basket

Contoh (composite)

JadwalKuliah

KodeKul	NamaKul	Dosen	Kelas	Jadwal

Dimana nilai atribut jadwal digabung hari dan jam

Jika asumsi hari dan jam memegang peranan penting dalam sistem basis data,maka atribut jadwal perlu dipisah menjadi JadwalHari dan JadwalJam sbb:

JadwalKuliah

KodeKul	NamaKul	Dosen	Kelas	JadwalHari	JadwalJam

Bentuk Normal Tahap Ke dua(2nd Normal Form)

Bentuk normal 2NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk 1NF, dan semua atribut selain primary key,secara utuh memiliki Functional Dependency pada primary key

Sebuah tabel tidak memenuhi 2NF jika ada atribut yang ketergantungannya(Functional Dependency) hanya bersifat parsial saja(hanya tergantung pada sebagian dari primary key)

Jika terdapat atribut yang tidak memiliki ketergantungan terhadap primary key,maka atribut tersebut harus pindah atau dihilangkan

Contoh

Tabel berikut memenuhi 1NF tapi tidak termasuk 2NF:

Mhs_nrp	Mhs_nama	Mhs_alamat	Mk_kode	Mk_nama	Mk_sks	nihuruf

Tidak memenuhi 2NF,karena {Mhs_nrp,mk_kode} yang dianggap sebagai primary key sedangkan:

{mhs_nrp,mk_kode} $\not\rightarrow$ mhs_nama

{mhs_nrp,mk_kode} $\not\rightarrow$ mhs_alamat

{mhs_nrp,mk_kode} $\not\rightarrow$ mhs_nama

{mhs_nrp,mk_kode} $\not\rightarrow$ mhs_sks

{mhs_nrp,mk_kode} $\not\rightarrow$ nihuruf

Tabel diatas perlu didekomposisi menjadi beberapa tabel yang memenuhi syarat 2NF

Contoh

Functional Dependencynya sbb:

{mhs_nrp,mk_kode} $\rightarrow$ nihuruf (fd1)

Mhs_nrp $\rightarrow$ {mhs_nama,mhs_alamat} (fd2)

Mk_kode $\rightarrow$ {mk_nama,mk_sks} (fd3)

Fd1 (mhs_nrp,mk_kode,nihuruf) $\rightarrow$ Tabel Nilai

Fd2 (Mhs_nrp,mhs_nama,mhs_alamat) $\rightarrow$ Tabel mahasiswa

Fd3 (mk_kode,mk_nama,mk_sks) $\rightarrow$ Tabel MataKuliah

Bentuk Normal Tahap Ketiga (3rd Normal Form/3NF)

Bentuk normal 3NF terpenuhi jika telah memenuhi 2NF,jika tidak ada atribut *non primary key* yang memiliki ketergantungan terhadap atribut *non primary key* yang lainnya

Untuk setiap Functional Dependency dengan notasi X $\rightarrow$ A,maka;

- X harus menjadi superkey pada tabel tsb
- Atau A merupakan bagian dari primary key pada tabel tsb

Contoh

Tabel berikut memenuhi 2NF,tapi tidak memenuhi 3NF:

Mahasiswa

Nrp	Nama	Alm_Jalan	Alm_Kota	Alm_Provinsi	Alm_Kodepos

Karena masih ada atribut *non primary key* (yakni alm_kota dan alm_provinsi) yang memiliki ketergantungan terhadap atribut *non primary key* yang lain (yakni alm_kodepos)

Alm_kodepos ⟶ {alm_provinsi,alm_kota}

Sehingga tabel tersebut perlu didekomposisi menjadi

Mahasiswa(Nrp,nama,alm_jalan,alm_kodepos)

Kodepos(alm_kodepos,alm_provinsi,alm_kota)

BCNF

Bentuk BCNF terpenuhi dalam sebuah tabel,jika untuk setiap *functional dependency* terhadap setiap atribut atau gabungan atribut dalam bentuk:X ⟶ Y maka X adalah *superkey*

Tabel tersebut harus didekomposisi berdasarkan *functional dependency* yang ada,sehingga X menjadi super key dari tabel-tabel hasil dekomposisi

Setiap tabel dalam BCNF merupakan 3NF.Akan tetapi setiap 3NF belum tentu termasuk BCNF.

Perbedaanya,untuk functional dependency X ⟶ A,BCNF tidak membolehkan A sebagai bagian dari primary key

Bentuk Normal Tahap Keempat (4th Normal Form/4NF)

Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF,dan tabel tersebut tidak boleh memiliki lebih dari sebuah *multivalued atribute*

Untuk setiap *multivalued dependencies* (MVD) juga harus merupakan *functional dependencies*

Contoh

Misal,tabel berikut tidak memenuhi 4NF:

Employee	Project	Skill
jim	11	design
mary	5	analys
mary	null	penyiaran

Setiap employee dapat bekerja di lebih dari project dan dapat memiliki lebih dari satu skill.Untuk kasus seperti ini tabel tersebut harus didekomposisi menjadi

(Employee,project)

(employee,skill)

Bentuk Normal Tahap Keempat(5th Normal Form/5NF)

Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel-tabel yang lebih kecil

Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependency,*5NF dibentuk berdasarkan konsep *join dependence*.Yakni apabila sebuah tabel telah didekomposisi menjadi tabel-tabel lebih kecil,harus bisa digabungkan lagi(join) untuk membentuk tabel semula

Binary Search

Sebuah algoritma pencrian biner(atau pemilahan biner) adalah sebuah teknik untuk menemukan nilai tertentu dalam sebuah larik(array) linear,dengan menghilangkan *setengah data pada setiap langkah,dipakai secara luas tetapi tidak secara ekslusif dalam* ilmu komputer.

Sebuah pencarian biner mencari nilai tengah(median),melakukan sebuah pembandingan untuk menentukan apakah nilai yang dicari ada sebelum atau sesudahnya,kemudian mencari setengah sisanya dengan cara yang sama.Pada

intinya,algoritma ini menggunakan prinsip divide and conquer,dimana sebuah masalah atau tujuan diselesaikan dengan cara mempartisi masalah menjadi bagian yang lebih kecil.Algoritma ini membagi sebuah tabel menjadi dua dan memproses satu bagian dari tabel itu saja.Algoritma ini bekerja dengan cara memilih record dengan indeks tengah dari tabel dan membandingkannya dengan record yang hendak dicari.Jika record tersebut lebih rendah atau lebih tinggi,maka tabel tersebut dibagi dua dan bagian tabel yang bersesuaian akan diproses kembali secara rekursif

Penerapan terbanyak dari pencarian biner adalah untuk mencari sebuah nilai tertentu dalam sebuah list terurut.Jika dibayangkan,pencarian biner dapat dilihat sebagai sebuah permainan tebak-tebakan,kita menebak sebuah bilanga,atau nomor tempat,dari daftar(*list)* nilai.

Pencarian diawali dengan memeriksa nilai yang ada pada posisi tengah lis; Oleh karena nilai-nilainya terurut,kita mengetahui apakah nilai terletak sebelum atau sesudah nilai yang ditengah tersebut,dan pencarian selanjutnya dilakukan terhadap setengah bagian dengan cara yang sama

Prinsip dari pencarian biner dapat dijelaskan sebagai berikut:

Data diambil dari posisi 1 sampai posisi akhir N

Kemudian cari posisi data tengah dengan rumus (posisi awal+posisi akhir)/2

Kemudian data yang dicari dibandingkan dengan data yang ditengah,apakah sama atau lebih kecil,atau lebih besar?

Jika lebih besar,maka proses pencarian dicari dengan posisi awal adalah posisi tengah+1

Jika lebih kecil,maka proses pencarian yang dicari dengan posisi akhir adalah posisi tengah-1

Jika data sama berarti ketemu

Contoh

Data yang dimasukan

12	15	57	8	37	2	25	5

1 2 3 4 5 6 7 8

Data yang telah terurut secara ascending

2	5	8	12	15	25	37	57

Kasus 1:cari=12

Probe Pertama :Tengah=(BatasAtas+BatasBawah)div 2=(1+8)=4

A[Tengah]=A[4]=12,berarti probe pertama data langsung ditemukan

Kasus 2 :cari=15

Probe pertama :Tengah=(BatasAtas+BatasBawah)div 2=(1+8)div 2=4

A[Tengah]=A[4]=12<cari=15,berarti BatasAtas=Tengah+1=4+1=5

Probe kedua :Tengah=(BatasAtas+BatasBawah)div 2=(5+8)div 2=6

A[Tengah]=A[6]=25>cari=15,berarti BatasBawah=Tengah-1=6-1=5

Probe ketiga :Tengah=(BatasAtas+BatasBawah)div 2=(5+5)div 2=5

A[Tengah]=A[5]=15<cari=15,berarti setelah probe ketiga,data ditemukan

Kasus 3 :cari=10

Kasus 2 :cari=15

Probe pertama :Tengah=(BatasAtas+BatasBawah)div 2=(1+8)div 2=4

A[Tengah]=A[4]=12>cari=10,berarti BatasBawah=Tengah-1=4-1=3

Probe kedua :Tengah=(BatasAtas+BatasBawah)div 2=(1+3)div 2=2

A[Tengah]=A[2]=5<cari=10,berarti BatasAtas=Tengah+1=2+1=3

Probe ketiga :Tengah=(BatasAtas+BatasBawah)div 2=(3+3)div 2=3

A[Tengah]=A[3]=8<cari=15,berarti setelah probe ketiga,data tidak

Ditemukan

Algoritma Binary Search

Algoritma pencarian biner dapat dituliskan sebagai berikut:

1. L ← 0
2. R ← N-1
3. Ketemu ←false
4. Selama (L<=R) dan(tidak ketemu) kerjakan baris 5 sampai dengan 8
5. m ←(L+R)/2

   83
6. Jika (Data[m]=x)maka ketemu ← true
7. Jika(x<Data[m] maka R ← m-1
8. Jika(x>Data[m])maka L ← m+1
9. Jika(ketemu) maka m adalah indeks dari data yang dicari,jika tidak data tidak ditemukan

Contoh

Sebuah aksi pencarian biner adalah sebuah permainan tebak-tebakan dimana seorang pemain harus menebak sebuah bilangan bulat positif yang dipilih oleh pemain lain diantara 1 dan N,dengan memanfaatkan jawaban pertanyaan berupa ya dan tidak.Misalnya N adalah 16 dan angka yang dipilih adalah 11,permainan dapat berjalan adalah sebagai berikut:

- Apakah angka lebih besar dari 8?(ya)
- Apakah angka lebih besar dari 12?(tidak)
- Apakah angka lebih besar dari 10?(ya)
- Apakah angka lebih besar dari 11?(tidak)

Sehingga angka tersebut pasti 11.pada setiap langkah,kita memilih sebuah angka yang tepat berada di tengah-tengah jangkauan nilai-nilai yang mungkin.Sebagai contoh,saat mengetahui angka tersebut lebih besar dari 8,tetapi lebih kecil atau sama dengan 12,kita

mengetahui untuk memilih angka ditengah-tengah jangkauan[9,12](pada kasus ini 10 adalah yang optimal)

Source Code

```
package binarysearch;

/**
 *
 * @author acer
 */
public class BinarySearch {
    public static int[] data =null;
    public static int awal,tengah,akhir,temp,count;

    /**
     * @param args the command line arguments
     */
```

```
    public static void main(String[] args) {
        // TODO code application logic here
        JTextArea area=new JTextArea();
        input1=JOptionPane.showInputDialog("Masukan Jumlah Data:");
        int jumlah=Integer.parseInt(input1);
        data=new int[jumlah];
        for(int x=0;x<data.length;x++)
        {
            data[x]=Integer.parseInt(JOptionPane.showInputDialog("masukan angka:"));
        }
            area.setText("\ndata yang dimasukan adalah:");

            for(int x=0;x<data.length;x++){
                area.append(data[x]+"");
            }
            JOptionPane.showMessageDialog(null,area,"                    Binary
search",JOptionPane.INFORMATION_MESSAGE);
            sorting();
            System.out.println();
            area.setText("hasil pengurutan adalah ");
            for(int x=0; x<data.length;x++){
                area.append(data[x]+"");
```

```
        }
        JOptionPane.showMessageDialog(null,area,"binary
search",JOptionPane.INFORMATION_MESSAGE);
        input2=JOptionPane.showInputDialog("masukkan data yang dicari");
        int cari=Integer.parseInt(input2);
        System.out.println();
        boolean temu=false;
        awal=0;
        akhir=data.length-1;
        temp=0;
        count=0;
        int iterasi=0;
        area.setText("probe\tAwal\tAkhir\tTengah\tNilai\n");
        while(temu!=true)
        {
           tengah=(awal+akhir)/2;
           iterasi++;
           if(data[tengah]==cari)
           {
              area.append(iterasi+"\t");
               area.append(awal+"\t");
                area.append(akhir+"\t");
```

```
        area.append(tengah+"\t");
         area.append(data[tengah]+"\n");
         temu=true;
         break;
}
else if(data[tengah]<cari)
{
    area.append(iterasi+"\t");
    area.append(awal+"\t");
      area.append(akhir+"\t");
       area.append(tengah+"\t");
        area.append(data[tengah]+"\n");
        akhir=tengah+1;
}
else if(data[tengah]>cari)
{
  area.append(iterasi+"\t");
    area.append(awal+"\t");
      area.append(akhir+"\t");
       area.append(tengah+"\t");
        area.append(data[tengah]+"\n");
```

```
                akhir=tengah-1;
            }
            if(temp!=data[tengah])
                temp=data[tengah];
            else
                count++;
            if(count==3)
                break;
        }
        JOptionPane.showMessageDialog(null,area,"binary
search",JOptionPane.INFORMATION_MESSAGE);
        if(temu==true){
            area.append("\n data"+cari+"ditemukan pada index ke"+tengah+"\n"+"dan
probe ke"+iterasi);
            JOptionPane.showMessageDialog(null,area,"binary
search",JOptionPane.INFORMATION_MESSAGE);
        }
        else
        {
            area.append("/n data"+cari+"tidak ditemukan");
            JOptionPane.showMessageDialog(null,area,"binary
serach",JOptionPane.INFORMATION_MESSAGE);
        }
        JOptionPane.showMessageDialog(null,"creat by ortega");
```

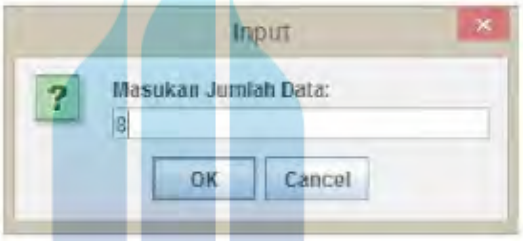
```
}
public static void sorting()
{
   int temp=0;
   for(int x=0;x,data.length;x++)
   {
      for(int y=0;y<data.length;y++)
      {
         if(x==y)
            continue;
         else
         {
            if(data[x]<data[y])
            {
               temp=data[y];
               data[y]=data[x];
               data[x]=temp;

            }
         }
      }
```

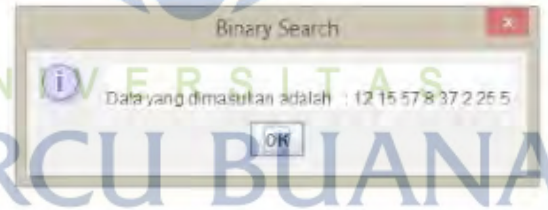
```
    }

    }
}
```

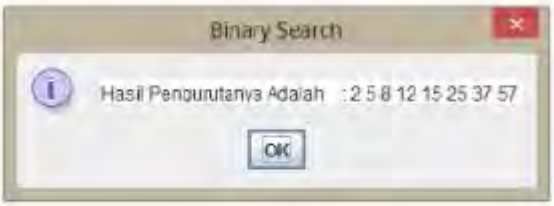
Output Program

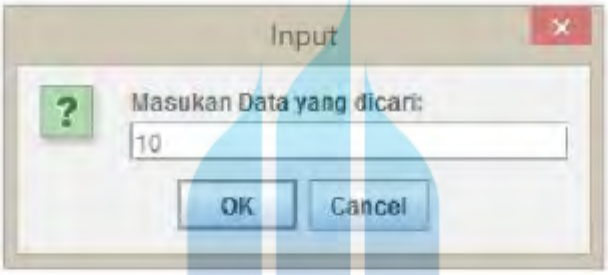
Apabila data yang dicari ditemukan:

Apabila data yang dicari tidak ditemukan:









Network Specialist

IP address dan subnetting

Subnet mask ip address

Ip address 192.168.100.80

Ip address 10.10.10.1

Kelas A=255.0.0.0

Kelas B=255.255.0.0

Kelas C=255.255.255.0

Ip 192.168.100.80/25

IP yang bisa digunakan 192.168.100.1-192.168.100.126

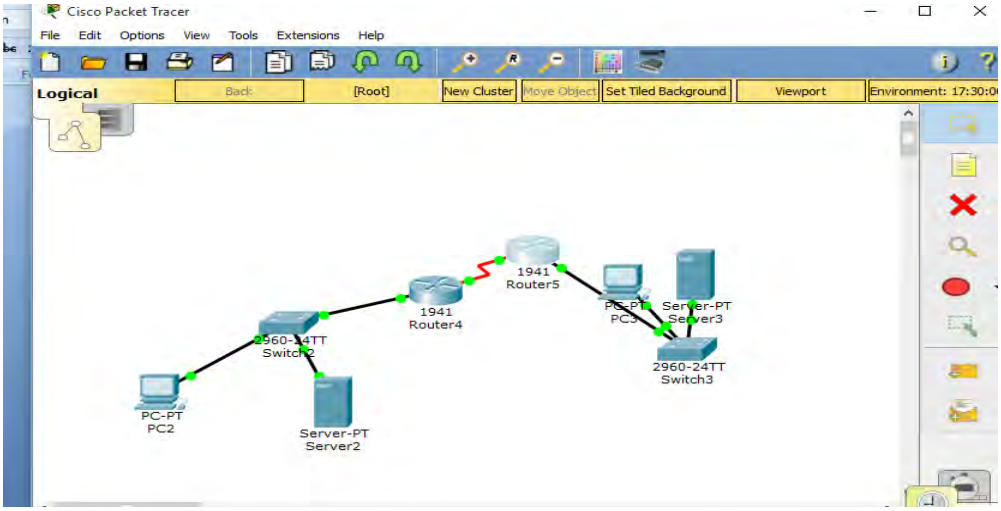
32-25=7; 7^2=128 kurangi 2 untuk network ID dan Broadcast

Host pertama 192.168.100.0

Host terakhir 192.168.100.127

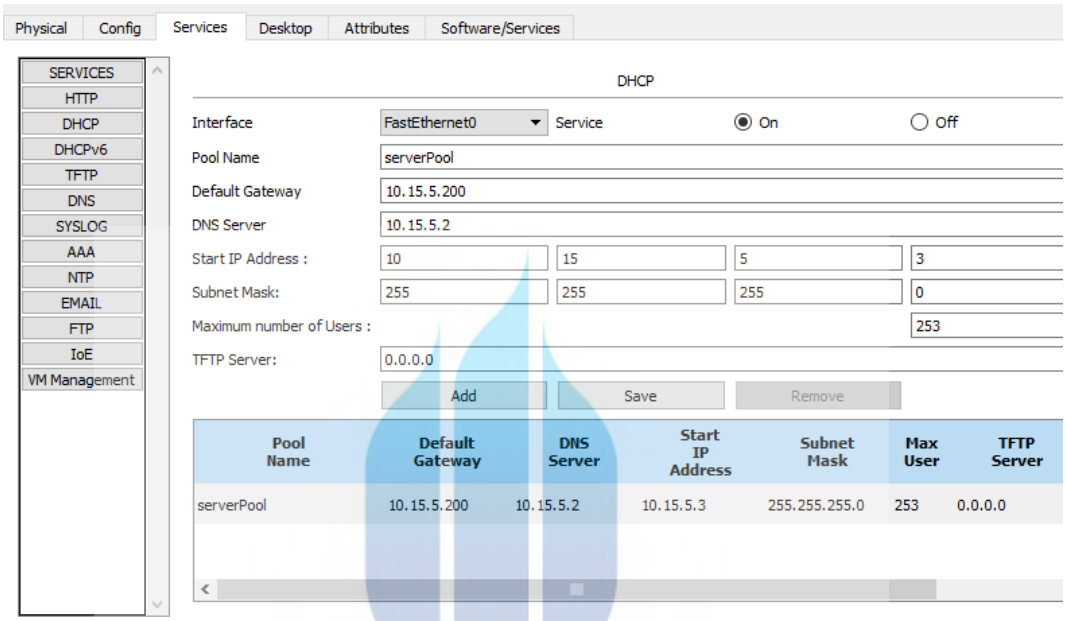
Subnet mask 255.255.255.128

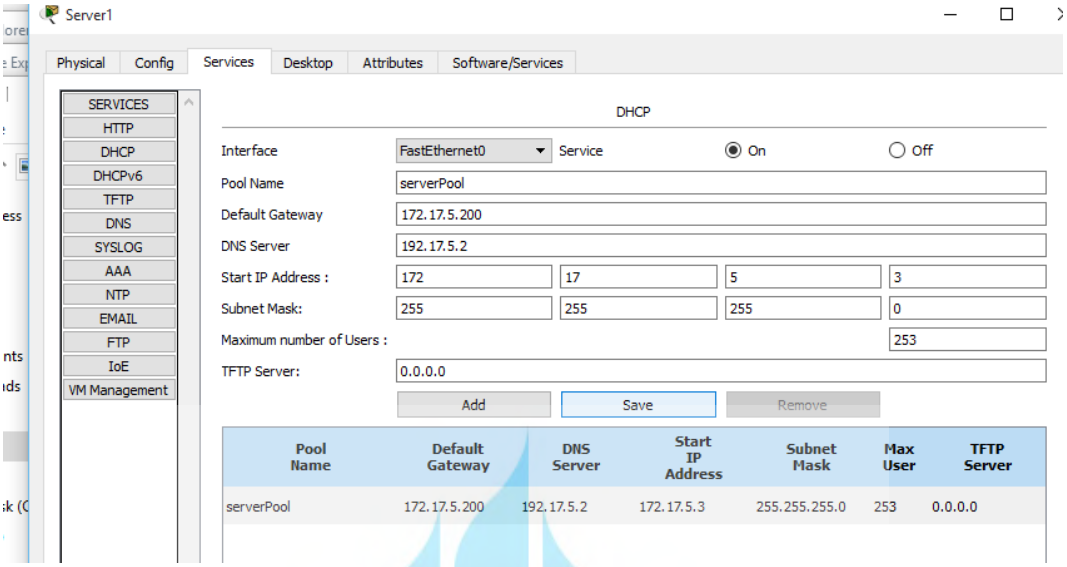
Gambar Jaringan



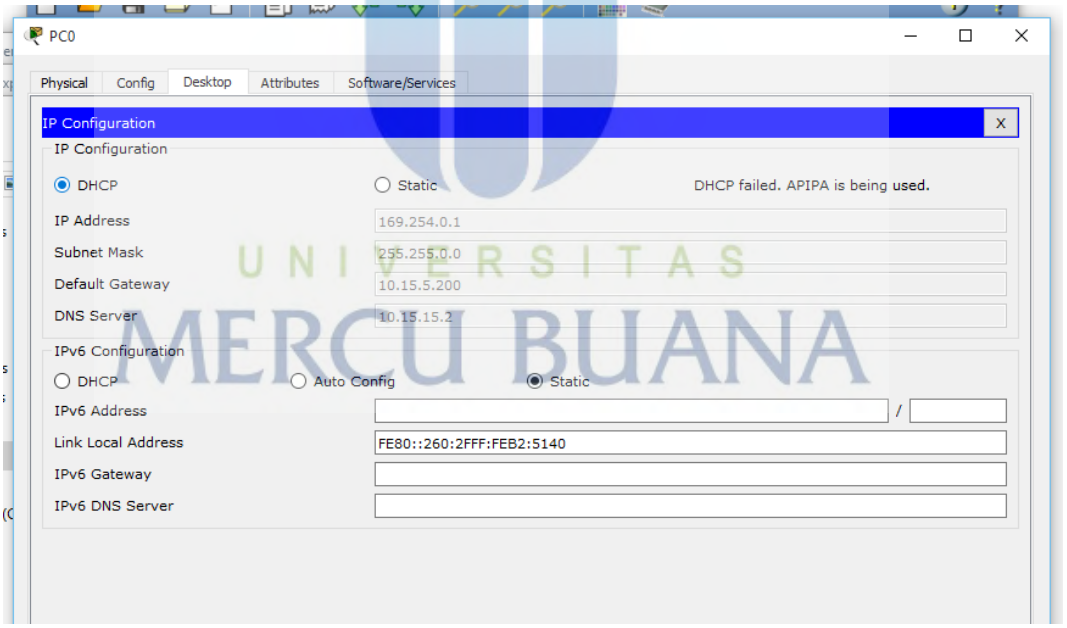
Setelah semua kabel terkoneksi, mensetting server masing-masing.dahulu mensetting server 0

Caranya klik pada server kemudian klik pada server pilih tab Dekstop lalu pilih IP Configuration.Kemudian langkah selanjutnya bisa dilihat pada gambar dibawah

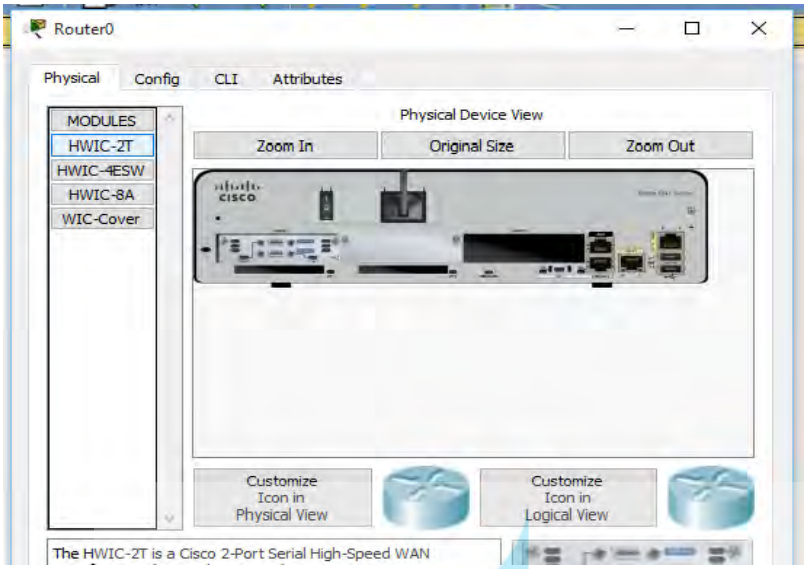




Setelah semua server dikonfigurasi,selanjutnya setting ip menjadi DHCP

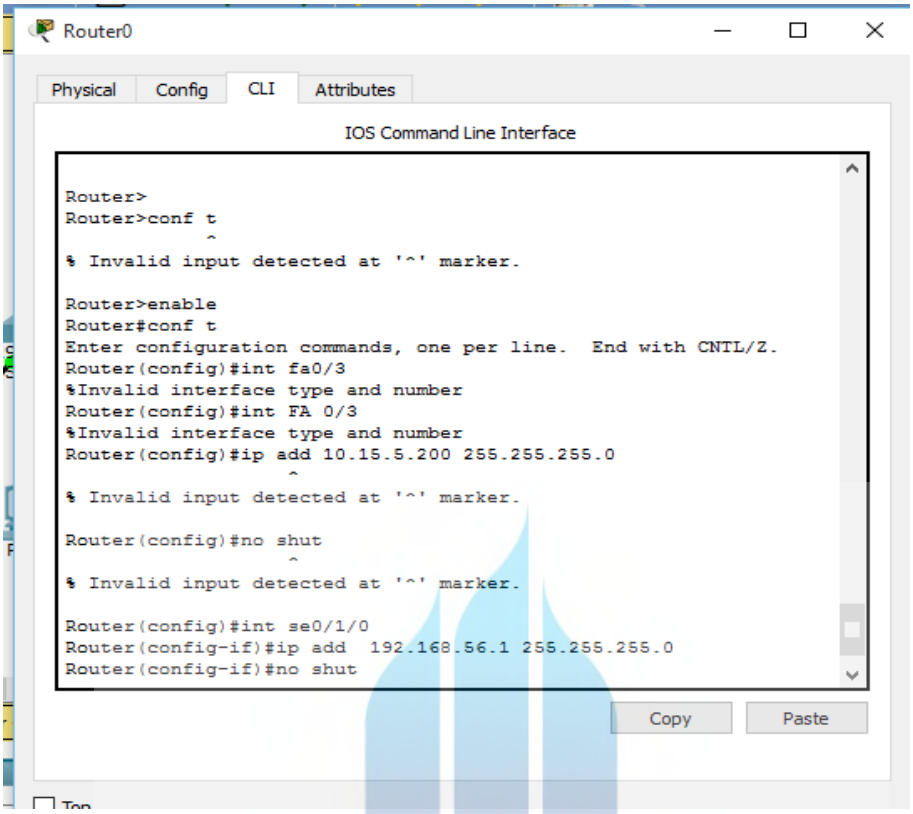


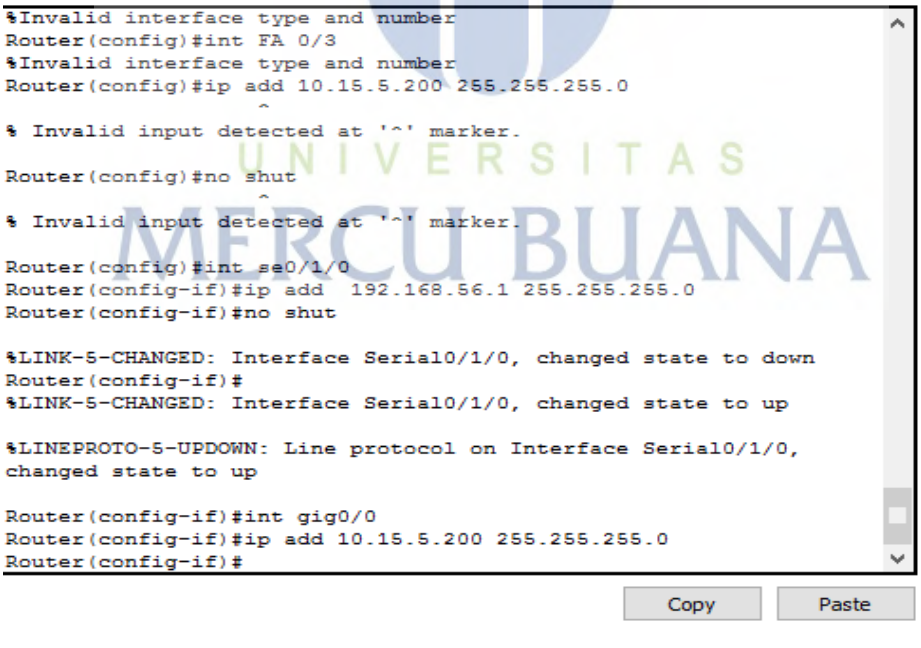
Lalu selanjutnya mengkonfigurasi router tapi sebelum melakukan konfigurasi,karena akan menggunakan kabel serial untuk menghubungkan router ke router maka kita harus menambahkan module terlebih dahulu pada routernya,caranya klik routernya dengan cara meng”drag” module tersebut pada router

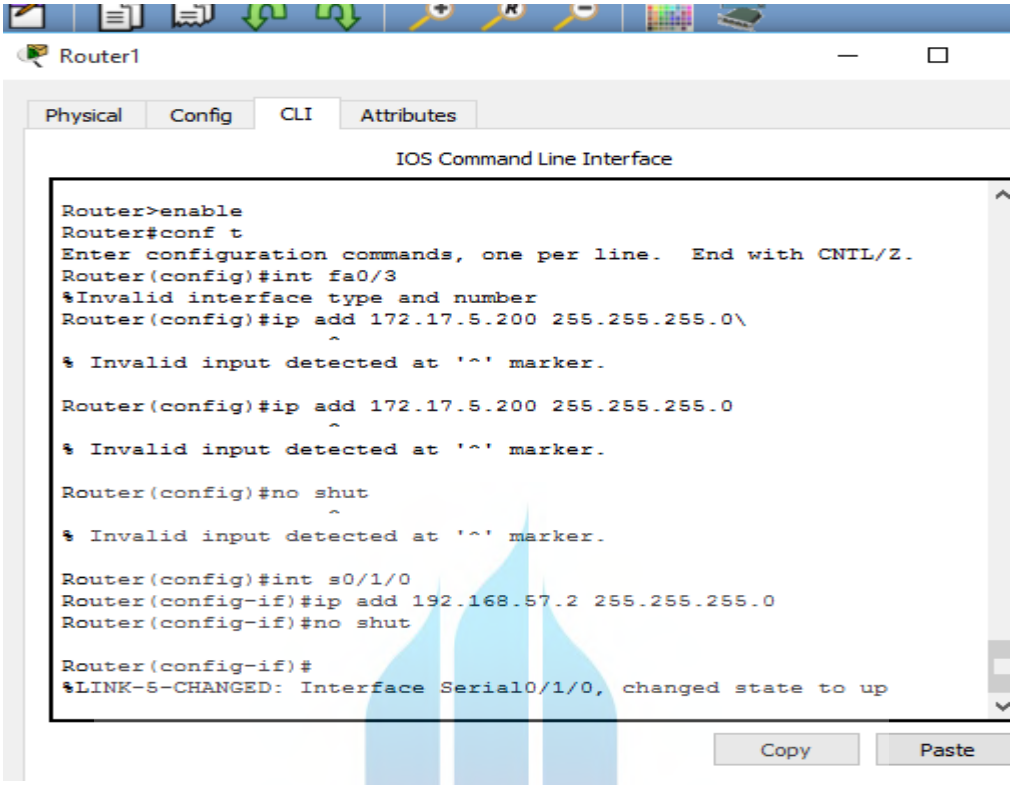


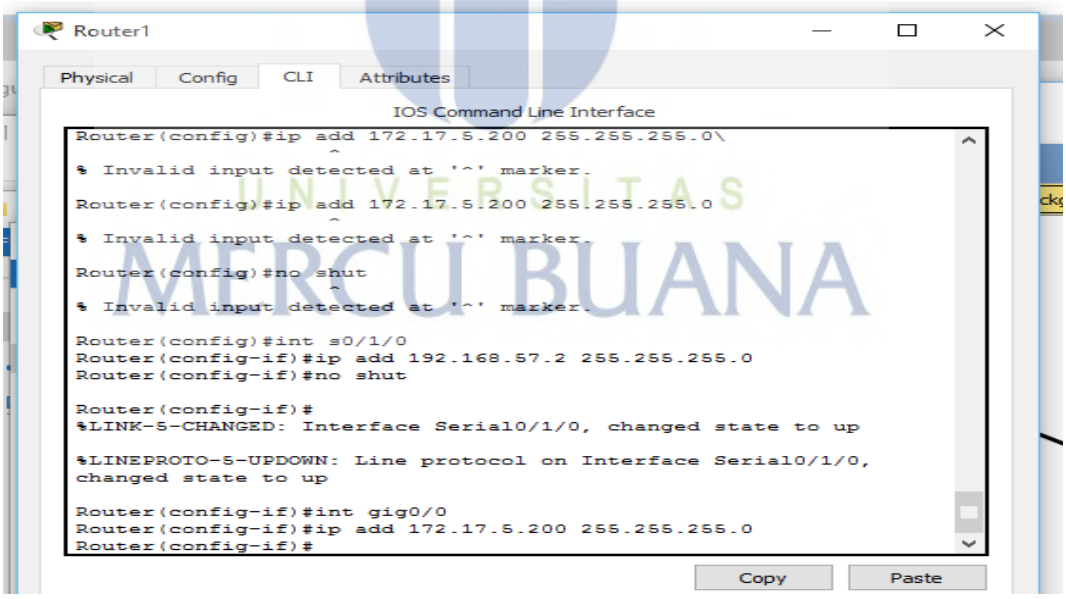
Lalu sambungkan masing-masing router ke switch dengan menggunakan kabel straight pada fa 0/3 dan sambungkan router 0 dengan router 1 dengan masing-masing menggunakan serial 0/1/0

Setelah semuanya dihubungkan,lalu selanjutnya kik router → pilih tab CLI → lalu masukan config seperti gambar dibawah.Untuk setting berawal dari router 0

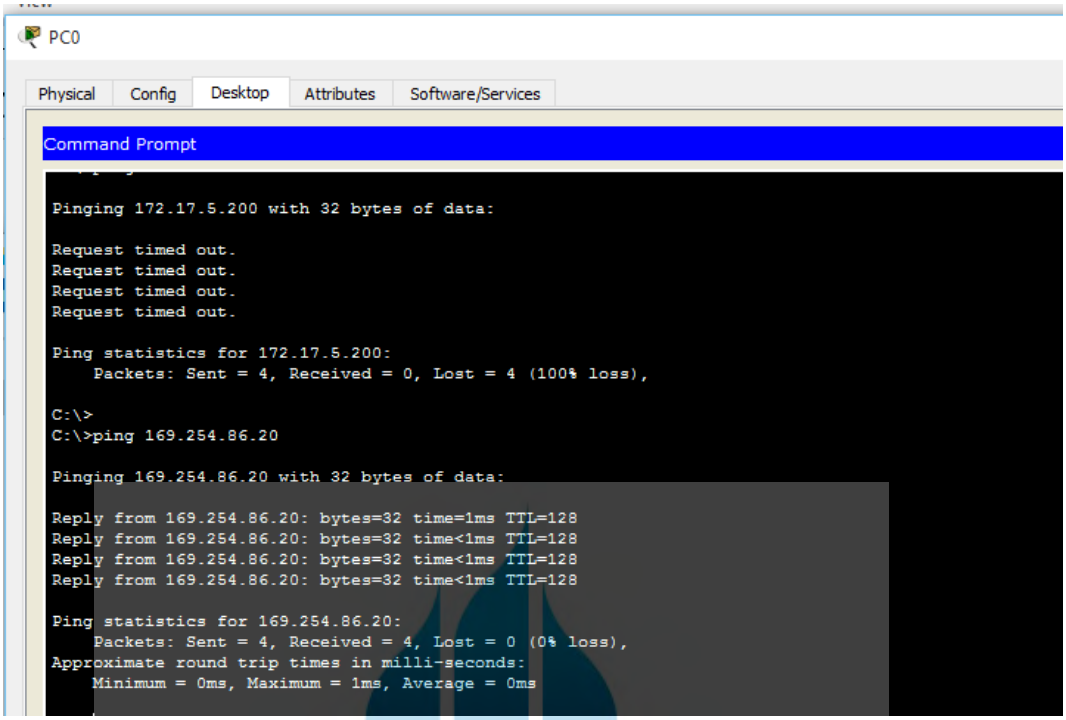








Dengan selesainya mengkonfigurasi router selanjutnya kita test koneksi apakah sudah terhubung dengan baik atau tidak.Dengan melakukan ping pc1 ke server0



MATERI UJI YANG DIPILIH

NO	Nama matakuliah	Ceklist
	Wajib	
1	Algoritma dan Struktur Data	
2	Rekayasa Perangkat Lunak	
	Pilihan Kompetensi	
1	Sistem Basis Data	
2	Network Specialist	

Model Portotype

Model Portoype adalah salah satu model sederhana pembuatan software yang dimana mengijinkan pengguna memiliki suatu gambaran awal/ dasar tentang program serta melakukan pengujian awal yang didasarkan pada konsep model kerja(working model)

Tahap-Tahapan

Selain itu,untuk memodelkan sebuah perangkat lunak dibutuhkan beberapa tahapan di dalam proses pengembangannya. Tahapan inilah yang akan menetukan keberhasilan dari sebuah software itu. Pengembang perangkat lunak harus memperhatikan tahapn dalam metode portotyping agar software finalnya dapat diterima oleh penggunanya. Dan tahapan-tahapan dalam portotyping tersebut adalah sebagai berikut:

1 Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan keseluruhan perangkat lunak,mengidenfinisikan semua kebutuhan,dan garis besar sistem yang akan dibuat

2 Membangun portotyping

Membangun portotyping dengan membuat perancangan sementara yang berpusat pada penyajian kepada pelanggan(misalnya dengan membuat input dan contoh outputnya)

3 Evaluasi portotyping

Evaluasi ini dilakukan oleh pelanggan apakah portotyping yang sudah dibangun sudah sesuai dengan keingina pelanggan.Jika sudah sesuai maka langkah keempat akan diambil.Jika tidak,maka portotyping diperbaiki dengan mengulang langkah 1,2,3

4 Mengkodekan system

Dalam tahap ini portotyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai

5 Menguji system

Setelah sistem sudah menjadi suatu perangkat lunak yang siap dipakai,harus dites sebelum digunakan.Pengujian ini dilakukan dengan White Box,Black Box,Basis Path,pengujian arsitektur dan lain-lain

6 Evaluasi Sistem

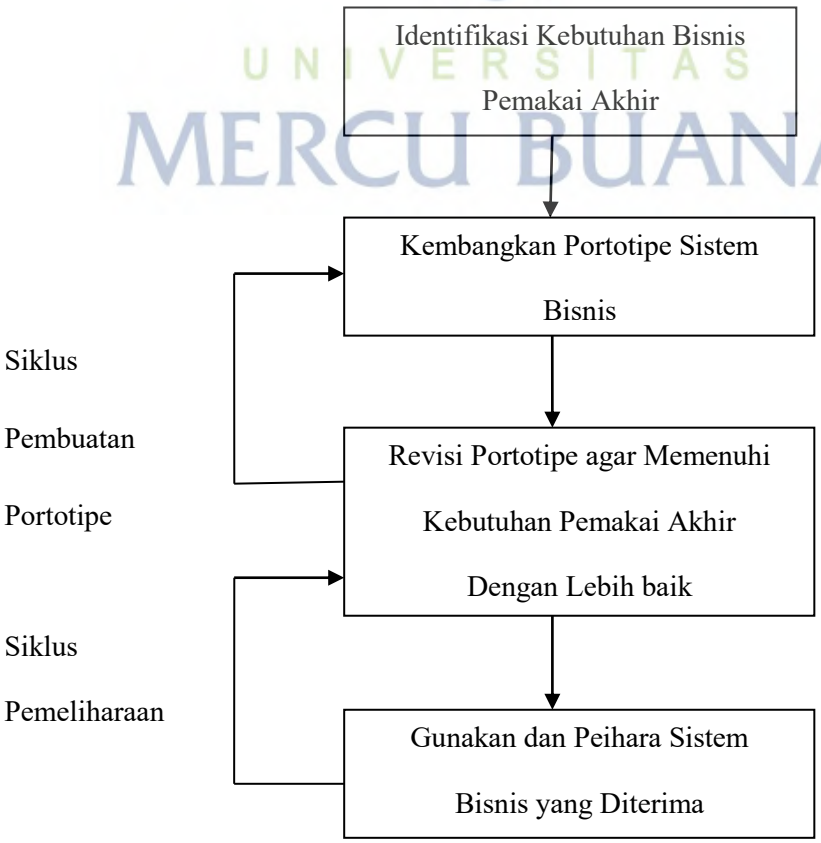
Pelanggan mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkan.Jika sudah,maka langkah ketujuh dilakukan,jika belum maka mengulangi langkah ke 4 dan 5

7 Menggunakan system

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan

Proses Pembuatan Portotipe

Proses pembuatan portotipe merupakan proses yang interaktif dan berulang-ulang yang menggabungkan langkah-langkah siklus pengembangan tradisional.Portotipe dievaluasi beberapa kali sebelum pemakai akhir menyatakan protipe tersebut diterima.Gambar dibawah ini megilustrasikan proses pembuatan portotipe:

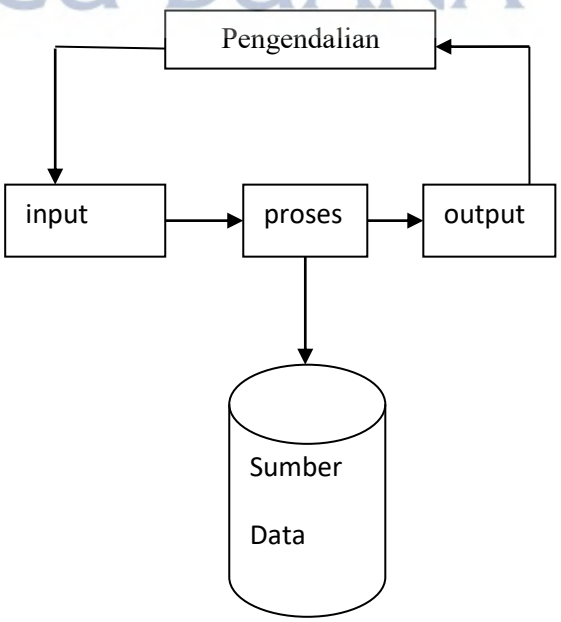


Langkah-Langkah Portotyping

1. Analisis Kebutuhan Sistem

Pembangunan sitem informasi memerlukan penyelidikan dan analisis mengenai alasan timbulnya ide atau gagasan untuk membangun dan mengembangkan sistem informasi.Analisis dilakukan untuk melihat berbaga komponen yang dipakai sistem yang sedang berjalan meliputi *hardware,software,*jaringan dan sumber daya manusia.Analis juga mendokumentasikan aktivitas sistem informasi meliputi *input,*pemrosesan,*output*,penyimpanan dan pengendalian(O'Brien,2005). Selanjutnya melakukan studi kelayakan(*Feasibility study)* untuk merumuskan informasi yang dibutuhkan pemakai akhir,kebutuhan sumber daya,biaya,manfaat dan kelayakan proyek yang diusulkan(Mulyanto,2009).Analisis kebutuhan sistem sebagai bagian dari studi awal bertujuan mengidentifikasi masalah dan kebutuhan spesifik sistem.Kebutuhan spesifik sistem adalah spesifikasi mengenai hal-hal yang akan dilakukan sistem ketika diimplementasikan(Mulyanto,2009) Analisis kebutuhan sistem harus mendenfinisikan kebutuhan sistem yang spesifik Antara lain:

1) Masukan yang diperlukan sistem(*input*)
2) Keluaran yang dihasilkan(*Output)*
3) Operasi-operasi yang dilakukan(proses)
4) Sumber data yang ditangani
5) Pengendlian (kontrol)



Spesifikasi Kebutuhan Sistem

Tahap analisis kebutuhan sistem memerlukan evaluasi untuk mengetahui kemampuan sistem dengan men definisikan apa yang seharusnya dapat dilakukan oleh sistem tersebut kemudian menentukan kriteria yang harus dipenuhi sistem. Beberapa kriteria yang harus dipenuhi adalah pencapaian tujuan,kecepatan,biaya,kualitas informasi yang dihasilkan,efisiensi san produktivitas,ketelitian dan validitas dan kehandalan atau realibilitas(Mulyanto,2009)

b. Desain Sistem

Analisis Sistem (*system analisis*) mendeskripsikan apa yang harus dilakukan sistem untuk memenuhi kebutuhan informasi pemakai.Desain sistem (*system design*) menentukan bagaimana sistem akan memenuhi tujuan tersebut.Desain sistem terdiri dari aktivitas desain yang menghasilkan spesifikasi fungsional.Desain sistem dapat dipandang sebagai desain *interface*,data dan proses dengan tujuan menghasilkan spesifikasi yang sesuai dengan produk dan metode *interface* pemakai,struktur database serta pemrosesan dan prosedur pengendalian(loanna et al.,2007).

Desain sistem akan menghasilkan paket *software* portotipe,produk yang baik sebaiknya mencakup tujuh bagian:

1) Fitur menu yang cepat dan mudah
2) Tampilan input dan output
3) Laporan yang mudah dicetak
4) Data *dictionary* yang menyimpan informasi pada setiap *field* termasuk panjang field,pengeditan dalam setiap laporan dan format *field* yang digunakan.
5) Database dengan format dan kunci *record* yag optimal
6) Menampilkan query *online* secara tepat ke data yang tersimpan pada *database*
7) Struktur yang sederhana dengan bahasa pemrograman yang mengizinkan pemakai melakukan pemrosesan khusus,waktu kejadian,prosedur otomatis dan lain-lain

c. Pengujian Sistem

Paket *software* portotipe diuji,diimplementasikan,dievaluasi dan dimodifikasi berulang-ulang higga dapat diterima pemakainya(O'Brien,2005).Pengujian sistem bertujuan menemukan kesalahan-kesalahan yang terjadi pada sistemdan melakukan revisi sistem.Tahap ini penting untuk memastikan bahwa sistem bebas dari kesalahan(Mulyanto,2009)

Menurut Sommerville(2001) pengujian sistem terdiri dari:

1) Pengujian unit untuk menguji komponen individual secara independen tanpa komponen sistem yang lain untuk menjamin sistem operasi yang benar

2) Pengujian modul yang terdiri dari komponen yang saling berhubungan
3) Pengujian sub sistem yang terdiri dari beberapa modul yang telah diintegrasikan
4) Pengujian sistem untuk menemukan kesalahan yang diakibatkan dari interaksi antara subsistem dengan interfacenya serta memvalidasi persyaratan fungsional dan non fungsioanal
5) Pengujian penerimaan dengan data yang di *entry* oleh pemakai dan bukan uji data simulasi
6) Dokumentasi berupa pencatatan terhadap setiap langkah pekerjaan dari awal sampai akhir pembuatan program

Pengujian sistem informasi berbasis *web* dapat menggunakan teknik dan metode pengujian perangkat lunak tradisional.Pengujian aplikasi *web* meliputi pengujian tautan,pengujian *browser*,pengujian usabilitas,pengujian muatan,tegangan dan pengujian malar(Simarmata,2009)

Penerimaan pengguna(*user*) terhadap sistem dapat dievaluasi dengan mengukur kepuasan user terhadap sistem yang diujikan.Pengukuran kepuasan meliputi tampilan sistem,kesesuaian dengan kebutuhan user,kecepatan dan ketetapan sistem untuk menghasilkan informasi yang diinginkan user.Ada beberapa model pengukuran kepuasan user terhadap sistem,diantaranya adalah *Technology Acceptance Model(TAM),End User Computing(*EUC),*Satisfaction,Task Technology fit(*TTF) *Analysis* dan *Human Organizational Technology*(HOT) Fit Model.

Salah satu model pengukuran yang telah diterjemahkan ke dalam beberapa bahasa berbeda dan tidak menujukkan perbedaan hasil pengukuran yang signifikan adalah *End User Computing*(EUC) *Satisfaction.*Model ini menekankan kepuasan user terhadap aspek teknologi meliputi aspek isi,keakuratan,format,waktu dan kemudahan penggunaan sistem(Chin&Mathew,2000).

d.Implementasi

Setelah portotipe diterima maka pada tahap ini merupakan tahap implementasi sistem yang siap dioperasikan dan selanjutnya terjadi proses pembelajaran terhadap sistem baru dan membandingkannya dengan sistem lama,evaluasi secara teknis dan operasional serta interaksi pengguna,sistem dan teknologi informasi

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ioi;

import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import java.util.Arrays;

/**
 *
 * @author acer
 */
public class IOI {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
      int[] nilaiInteger = {1,5,9,4,8,7,4,1,5,10};
     char[] nilaiKarakter = {'d', 'o', 'g', 'n', 'i', 'a', 'b', 'h', 'e'};

     String tampilan;
     JTextArea areaTampilan;

     // Menampilkan elemen larik integer
     tampilan = "Menampilkan elemen larik integer :\n";
     for (int x = 0; x < nilaiInteger.length; x++)
       tampilan += nilaiInteger[x] + "   ";

     Arrays.sort(nilaiInteger);

     // Menampilkan elemen larik integer setelah diurutkan
     tampilan += "\nMenampilkan elemen larik integer setelah diurutkan :\n";
     for (int x = 0; x < nilaiInteger.length; x++)
       tampilan += nilaiInteger[x] + "   ";

     // Menampilkan elemen larik karakter
     tampilan += "\n\nMenampilkan elemen larik karakter :\n";
     for (int x = 0; x < nilaiKarakter.length; x++)
       tampilan += nilaiKarakter[x] + "   ";
```

```
      Arrays.sort(nilaiKarakter);

      // Menampilkan elemen larik karakter setelah diurutkan
      tampilan += "\nMenampilkan elemen larik karakter setelah diurutkan :\n";
      for (int x = 0; x < nilaiKarakter.length; x++)
        tampilan += nilaiKarakter[x] + "   ";

      areaTampilan = new JTextArea();
      areaTampilan.setText(tampilan);
      JOptionPane.showMessageDialog(null, areaTampilan, "Mengurutkan Elemen
Larik",
        JOptionPane.INFORMATION_MESSAGE);

      // Mengakhiri aplikasi
      System.exit(0);
    }
}
```

Kesimpulan

Materi ini mencakup semua materi kompre saya algorima struktur data normalisasi

Studi pustaka

Ebook algoritma struktur data

www.pdfdrive.com

www.slideshare.net