BAB III
PEMROGRAMAN KOMPUTER

3.1 SEKILAS BAHASA PEMROGRAMAN

Dewasa ini komputer digunakan di hampir semua bidang kehidupan manusia, mulai dari pendidikan, bisnis, sampai dengan permainan. Berbicara tentang komputer tidak lepas dari pemrograman komputer. Hal ini karena komputer pada dasarnya adalah mesin yang tidak bisa apa-apa. Kita harus memberikan serangkaian instruksi kepada komputer agar mesin pintar ini dapat memecahkan suatu masalah. Langkah-langkah yang kita lakukan dalam memberikan instruksi kepada komputer untuk memecahkan masalah inilah yang dinamakan pemrograman komputer.

Pada dasarnya komputer adalah mesin digital, artinya komputer hanya mengenal kondisi ada arus listrik (biasanya dilambangkan dengan 1) dan tidak ada arus listrik (biasanya dilambangkan dengan 0). Dengan kata lain, kita harus menggunakan sandi 0 dan 1 untuk melakukan pemrograman komputer. Bahasa ini disebut bahasa mesin.

Karena bahasa mesin sangat susah, maka muncul ide untuk melambangkan untaian sandi 0 dan 1 dengan singkatan kata yang lebih mudah dipahami manusia. Singkatan kata ini kemudian sering disebut *mnemonic code*. Bahasa pemrograman yang menggunakan singkatan kata ini disebut bahasa *assembly*.

Pemrograman dengan bahasa *assembly* dirasakan banyak orang masih terlalu sulit. Akhirnya dikembangkanlah suatu bahasa pemrograman yang lebih mudah dipahami. Bahasa pemrograman ini menggunakan kata-kata yang mudah dikenali oleh manusia. Bahasa pemrograman seperti ini disebut 3GL (*Third Generation Language*). Beberapa orang menyebut HLL (*High Level Language*). Ada banyak contoh 3GL, antara lain, BASIC, Pascal, Delphi, C, C++, Cobol, dan sebagainya.

Perkembangan bahasa pemrograman tidak sampai pada bahasa generasi ketiga saja (3GL). Ada generasi lanjutan dari bahasa pemrograman, yaitu 4GL (*Fourth Generation Language*). Bahasa ini banyak digunakan untuk mengembangkan aplikasai berbasis data (*database*). Salah satu contohnya adalah SQL (*Structured Query Language*). Pada bahasa ini perintah-perintah yang digunakan lebih manusiawi, misalnya "SELECT * Nama, Alamat, FROM Karyawan", untuk mengambil data Nama, Alamat, dari basis data Karyawan.

3.2 ALGORITMA PROGRAM

Algoritma adalah urutan langkah berhingga untuk memecahkan masalah logika atau matematika (*Microsoft Bookshelf, 1997*). Kata "Algoritma" diambil dari nama seorang matematikawan yang juga seorang astronom berkebangsaan Arab bernama Al-Khowarizmi yang hidup pada abad ke-19.

Didalam mempelajari algoritma selalu ada 3 faktor yang selalu diperhatikan, yaitu :

a. Efisiensi dan efektifitas dari algoritma. Efisiensi dan efektifitas dari algoritma sangat ditentukan oleh kecepatan operasi dan kapasitas memori yang diperlukan pada komputer, juga kemampuan perangkat keras komputer dalam menjalankan algoritma tersebut. Suatu algoritma dapat dikatakan efisien dan efektif jika diaplikasikan atau dituliskan dengan bahasa pemrograman tertentu pada komputer, akan membutuhkan waktu penyelesaian yang paling singkat dengan kebutuhan alokasi memori yang paling minimum. Algoritma yang berhubungan dengan persoalan-persoalan numeris yang kompleks kadang-kadang membutuhkan fasilitas yang lengkap dari perangkat komputer.misalnya komputer yang dilengkapi *Math-Co-processor* akan lebih baik dalam memecahkan persoalan numeris dibandingkan komputer biasa.
b. Bahasa pemrograman yang menunjang aplikasi algoritma pada komputer. Kadang-kadang dapat dijumpai suatu algoritma yang sangat sulit diaplikasikan pada komputer karena tidak adanya bahasa pemrograman yang menunjang pemakaian algoritma tersebut. Meskipun algoritma tersebut dikatakan efisien dan efektif tetapi tidak ditunjang fasilitas bahasa

pemrograman yang memadai maka eksekusi dari algoritma akan menjadi kurang baik. Atau dengan perkataan lain jika bahasa pemrograman yang dipakai tidak menyediakan fasilitas yang memadai untuk eksekusi suatu algoritma, maka algoritma tersebut bisa menjadi kurang efisien dan efektif.

c. Tipe Algoritma. Sering kita jumpai adanya kemungkinan lebih dari satu algoritma dalam memecahkan suatu persoalan. Algoritma mana yang akan dipilih tentunya algoritma yang paling efisien dan efektif dalam memecahkan persoalan tersebut. Sering pula kita jumpai algoritma yang efisien dan efektif untuk memecahkan suatu persoalan kadang menjadi kurang efektif dan efisien dalam memecahkan persoalan yang sama dengan data yang berbeda. Hal ini dapat terjadi pada saat terjadinya perubahan masukan terhadap algoritma tersebut. Suatu algoritma dapat efisien dan efektif untuk mengerjakan atau mengolah data yang kecil. Namun kadang-kadang terjadi sebaliknya, jika jumlah data membesar, maka algoritma itu menjadi kurang efisien dan kurang efektif.

Dalam menuliskan instruksi-instruksi yang terdapat dalam suatu algoritma dapat dilakukan dengan berbagai cara. Beberapa cara yang sering dilakukan adalah :

a. Mengikuti kaidah bahasa yang dipakai manusia.
b. Mengikuti kaidah bahasa yang dipakai didalam komputer.
c. Dengan menggunakan bagan alir (*flowchart*).

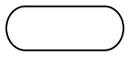
Jadi *flowchart* adalah suatu bagan yang menggambarkan urutan-urutan proses dalam menyelesaikan suatu persoalan. Semua persoalan ternyata dapat dipecahkan dengan membangun sekelompok atau block logika dari 3 (tiga) bentuk logika dasar, yaitu :

a. Urutan (*sequence*)
b. Pilihan (*selection*)
c. Pengulangan (*iteration*)

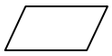
Pada pilihan maupun pengulangan selalu terdapat kondisi yang selalu diuji. Pada pengulangan selama kondisi benar maka akan mengerjakan proses yang sama (sekali atau berkali-kali) sedangkan jika kondisi salah akan mengerjakan proses yang lain. Sedangkan pada pilihan jika kondisi benar maka akan mengerjakan satu proses sedangkan kalau kondisi salah akan mengerjakan proses yang lain.

Ketiga logika dasar tersebut dapat digabungkan menjadi satu kesatuan logika yang lebih kompleks yang mencerminkan urutan-urutan proses dalam memecahkan suatu persoalan.

Dalam membuat bagan alir ada beberapa simbol yang sering dipakai, yaitu :



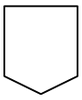
TERMINAL : dipakai untuk memulai dan mengakhiri flowchart.



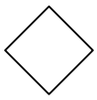
INPUT/OUTPUT : dipakai untuk membaca data dan menulis output.

PROSES : dipakai untuk mengolah data.

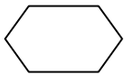
CONNECTOR (PENGHUBUNG) : dipakai untuk menghubungkan satu atau lebih bagan dalam satu halaman yang sama



OFF-PAGE CONNECTOR : dipakai untuk penghubung bagan antar halaman.



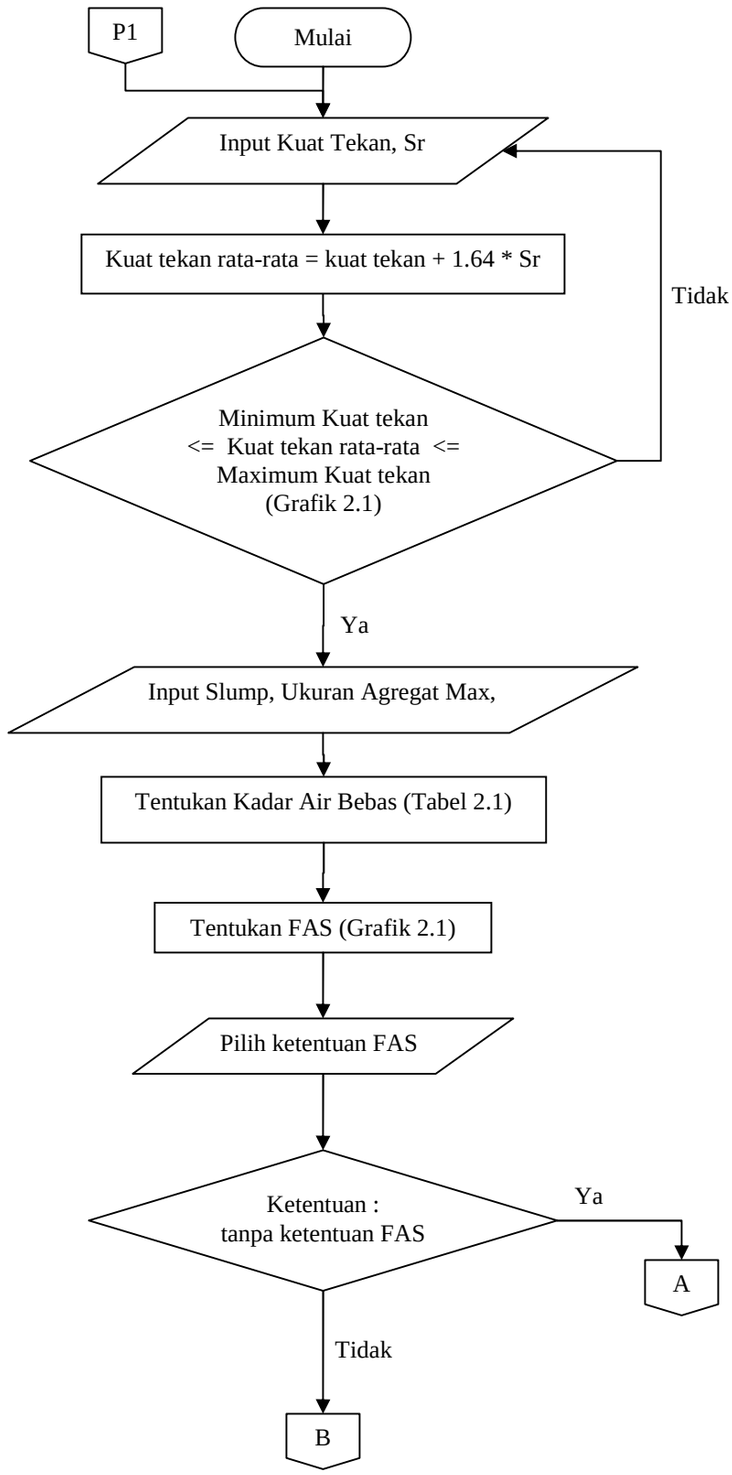
DECISION : dipakai untuk menentukan satu diantara dua kemungkinan pencabangan proses.

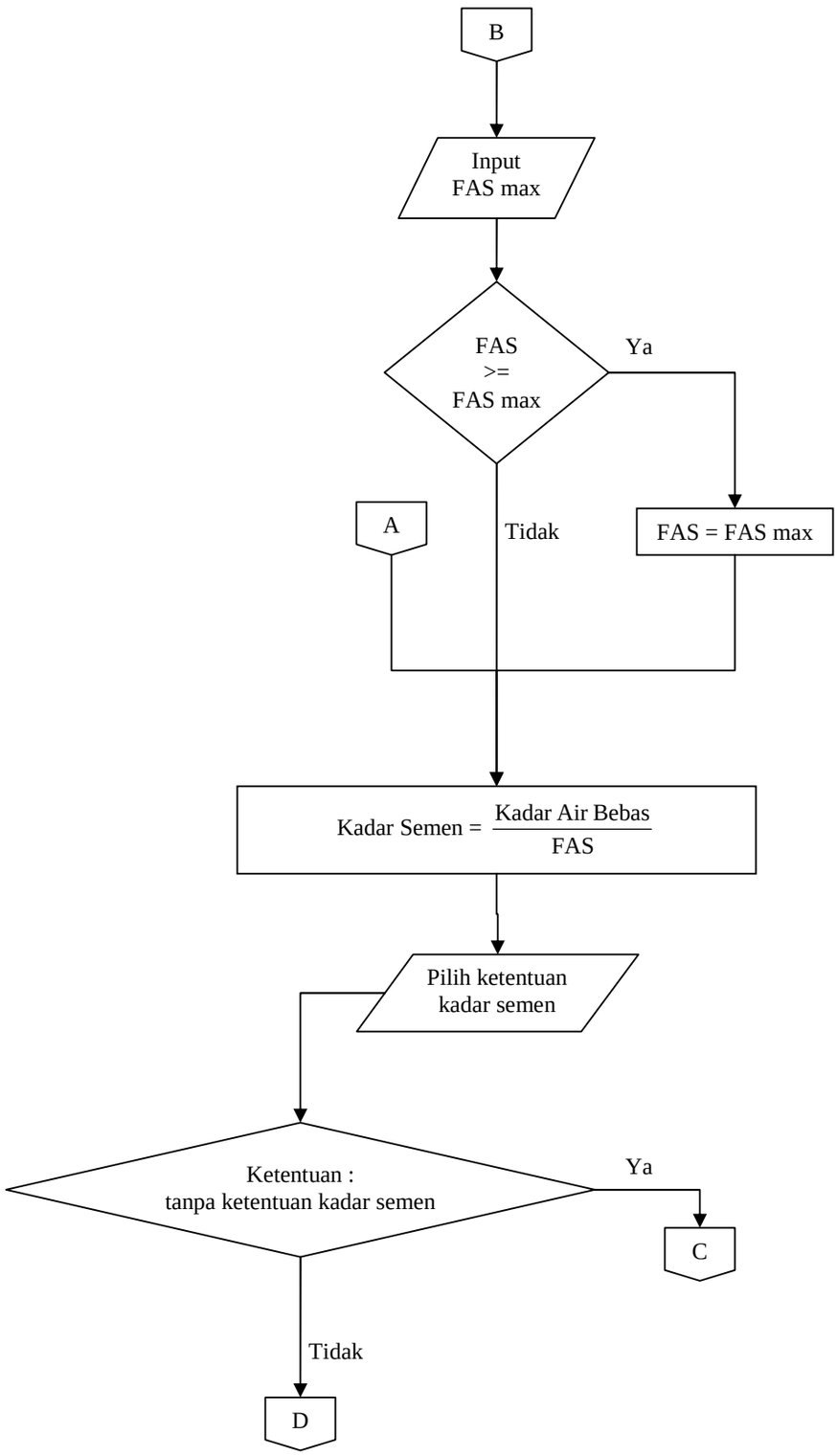


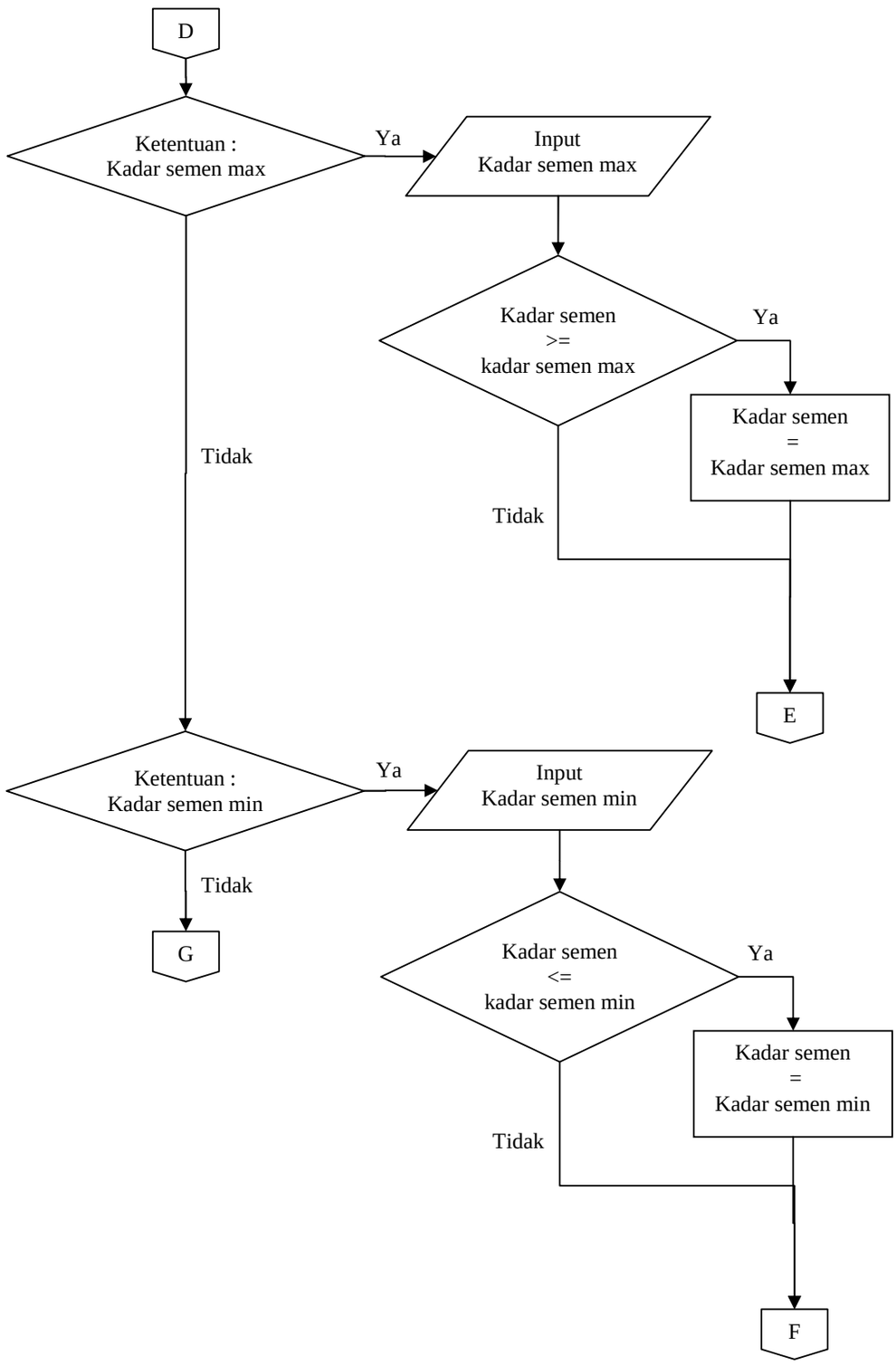
PREPARATION : dipakai untuk inisialisasi harga awal

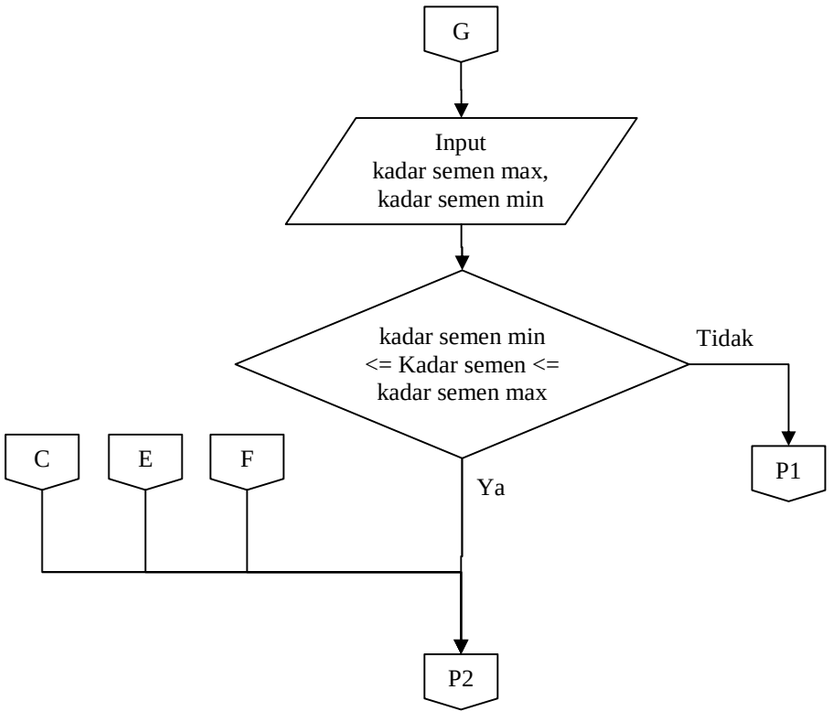
TANDA PANAH : dipakai untuk menunjukkan arah dari proses atau logika.

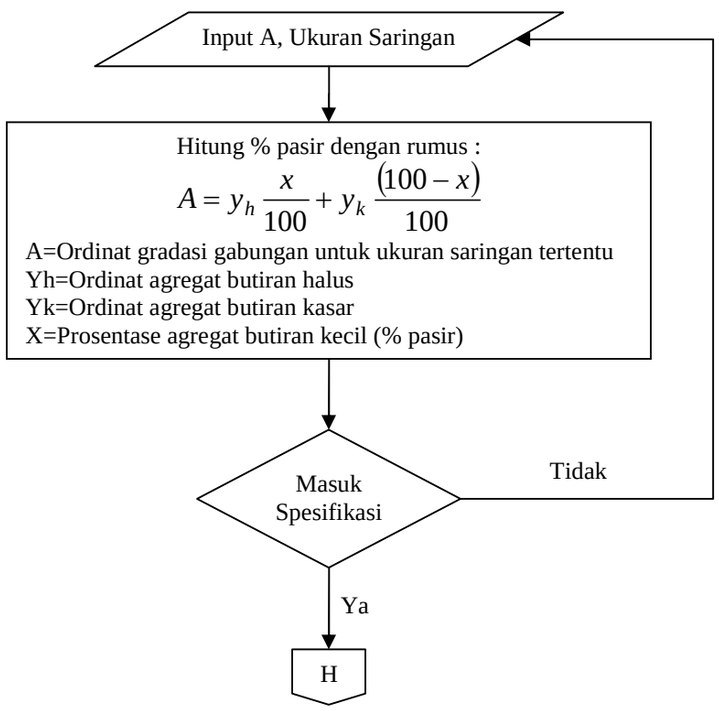
3.2.1 ALGORITMA METODE BRITISH STANDARD

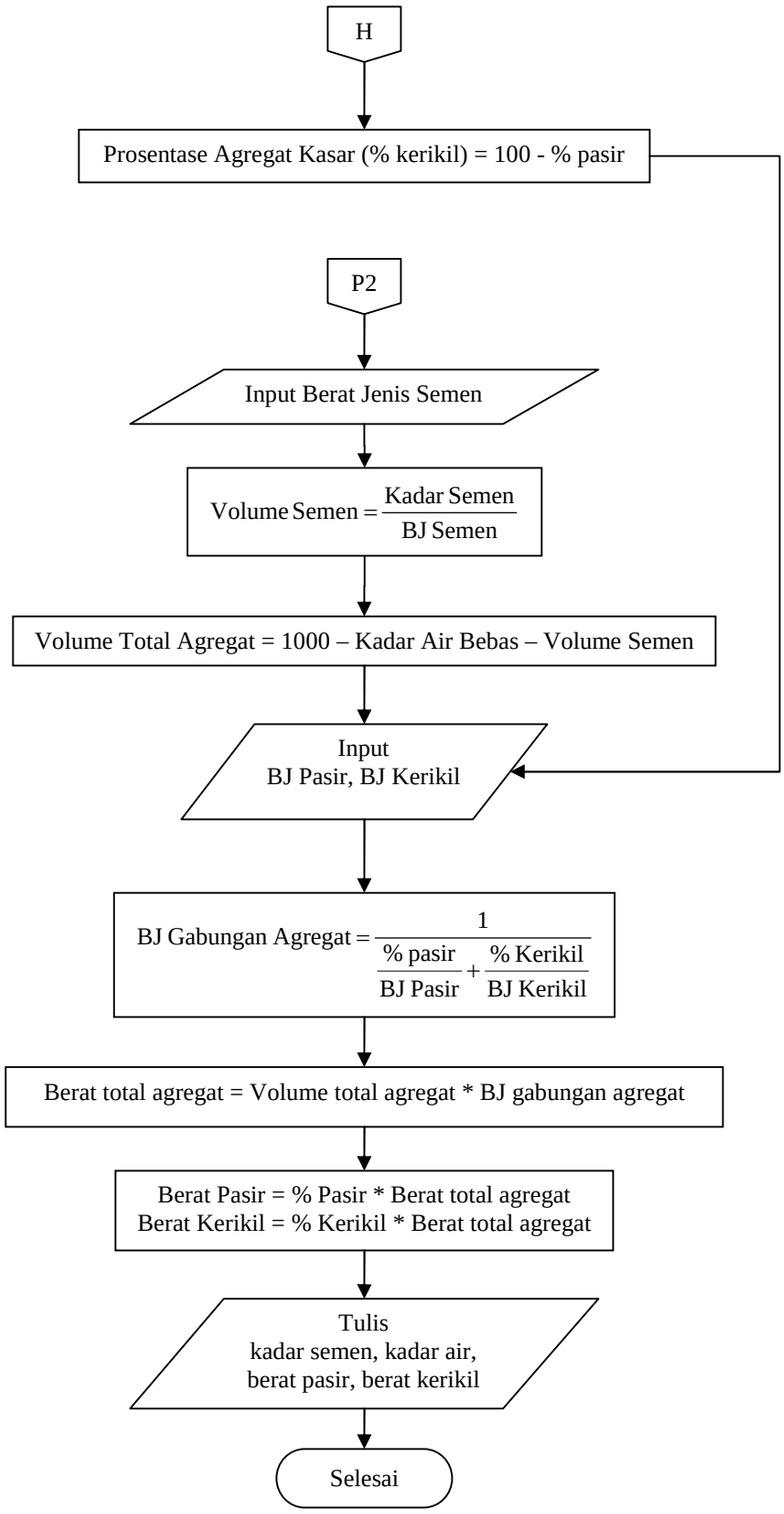



B
Input
FAS max
FAS
>=
FAS max
Ya
A
Tidak
FAS = FAS max
Kadar Semen = Kadar Air Bebas / FAS
Pilih ketentuan
kadar semen
Ketentuan :
tanpa ketentuan kadar semen
Ya
C
Tidak
D

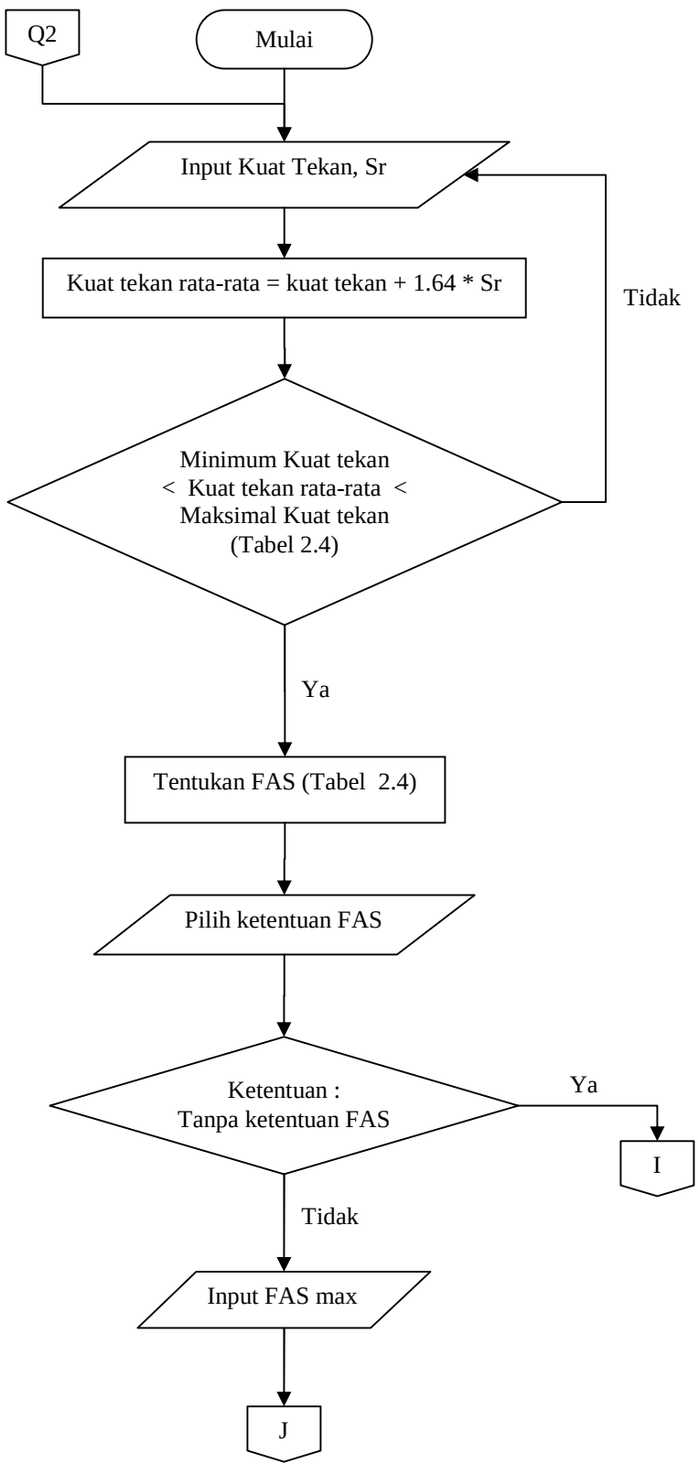

D
Ketentuan :
Kadar semen max
Ya
Input
Kadar semen max
Kadar semen
>=
kadar semen max
Ya
Kadar semen
=
Kadar semen max
Tidak
Tidak
E
Ketentuan :
Kadar semen min
Ya
Input
Kadar semen min
Tidak
G
Kadar semen
<=
kadar semen min
Ya
Kadar semen
=
Kadar semen min
Tidak
F

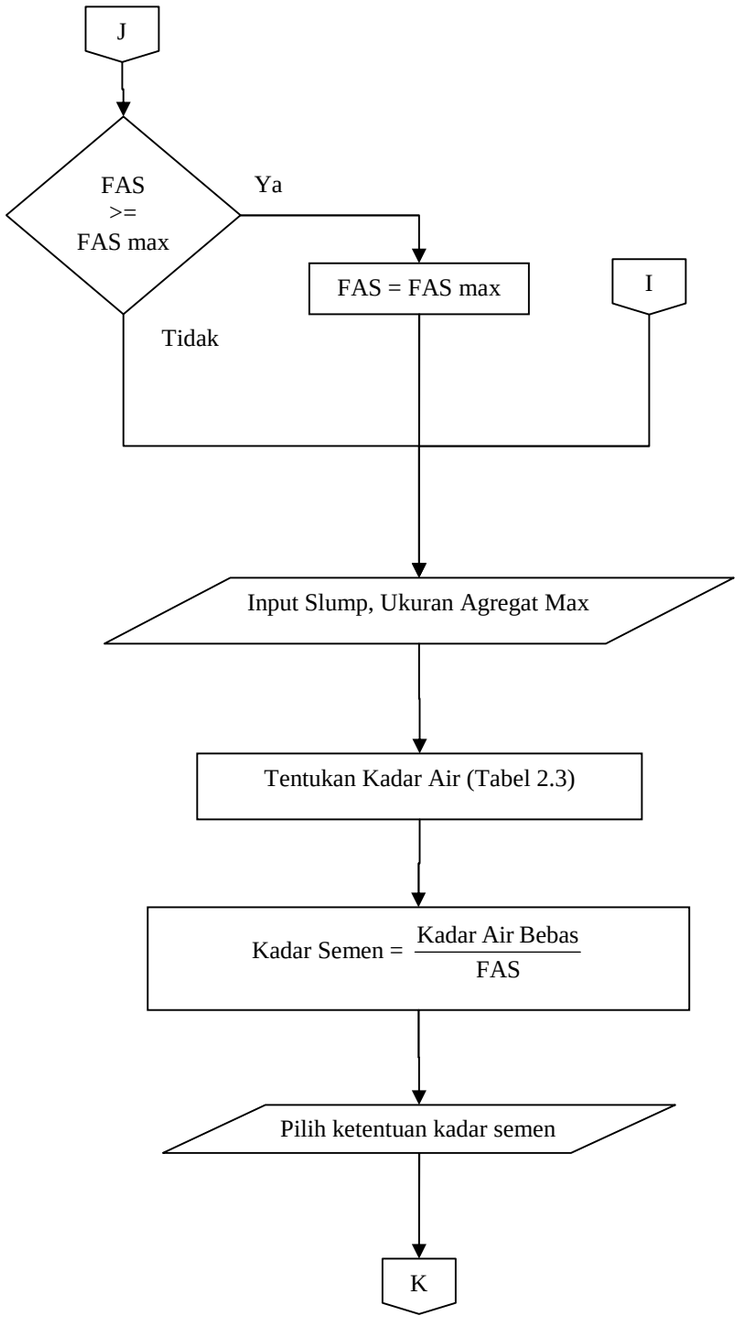

G
Input
kadar semen max,
kadar semen min
kadar semen min
<= Kadar semen <=
kadar semen max
Tidak
P1
C
E
F
Ya
P2

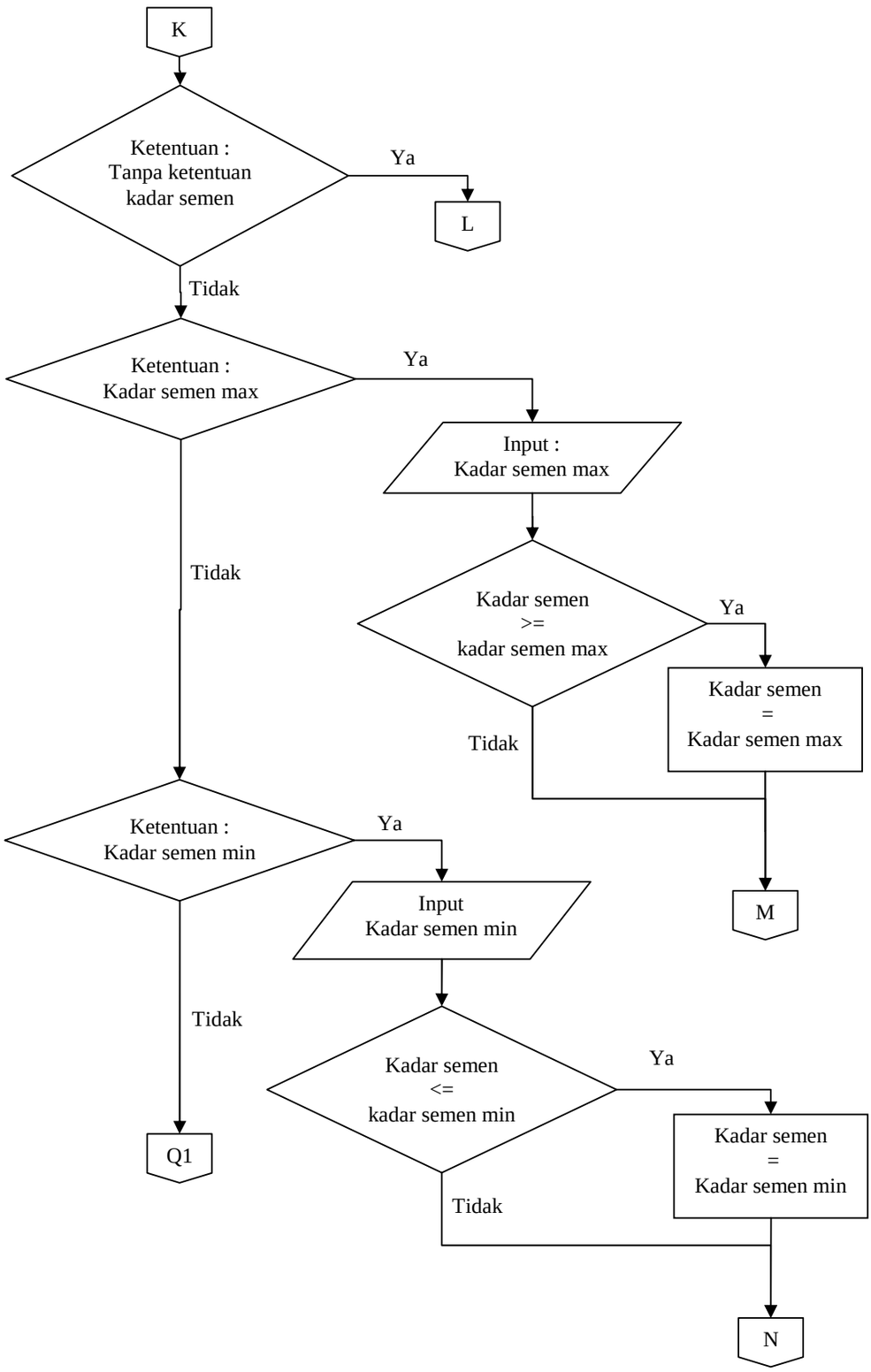

Input A, Ukuran Saringan
Hitung % pasir dengan rumus :
$A = y_h \frac{x}{100} + y_k \frac{(100 - x)}{100}$
A=Ordinat gradasi gabungan untuk ukuran saringan tertentu
Yh=Ordinat agregat butiran halus
Yk=Ordinat agregat butiran kasar
X=Prosentase agregat butiran kecil (% pasir)
Masuk
Spesifikasi
Tidak
Ya
H

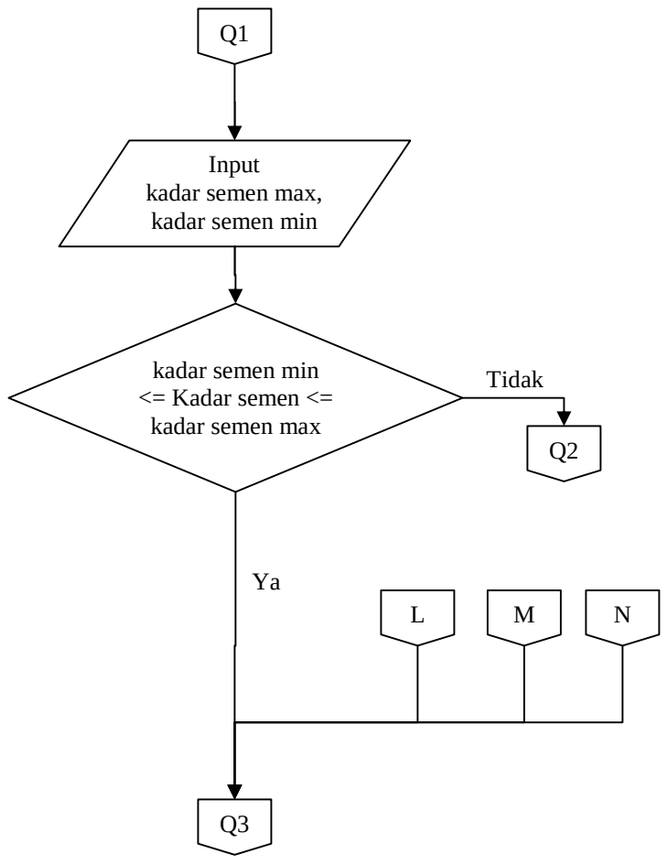

H
Prosentase Agregat Kasar (% kerikil) = 100 - % pasir
P2
Input Berat Jenis Semen
Volume Semen = Kadar Semen / BJ Semen
Volume Total Agregat = 1000 – Kadar Air Bebas – Volume Semen
Input
BJ Pasir, BJ Kerikil
BJ Gabungan Agregat = 1 / (% pasir / BJ Pasir + % Kerikil / BJ Kerikil)
Berat total agregat = Volume total agregat * BJ gabungan agregat
Berat Pasir = % Pasir * Berat total agregat
Berat Kerikil = % Kerikil * Berat total agregat
Tulis
kadar semen, kadar air,
berat pasir, berat kerikil
Selesai

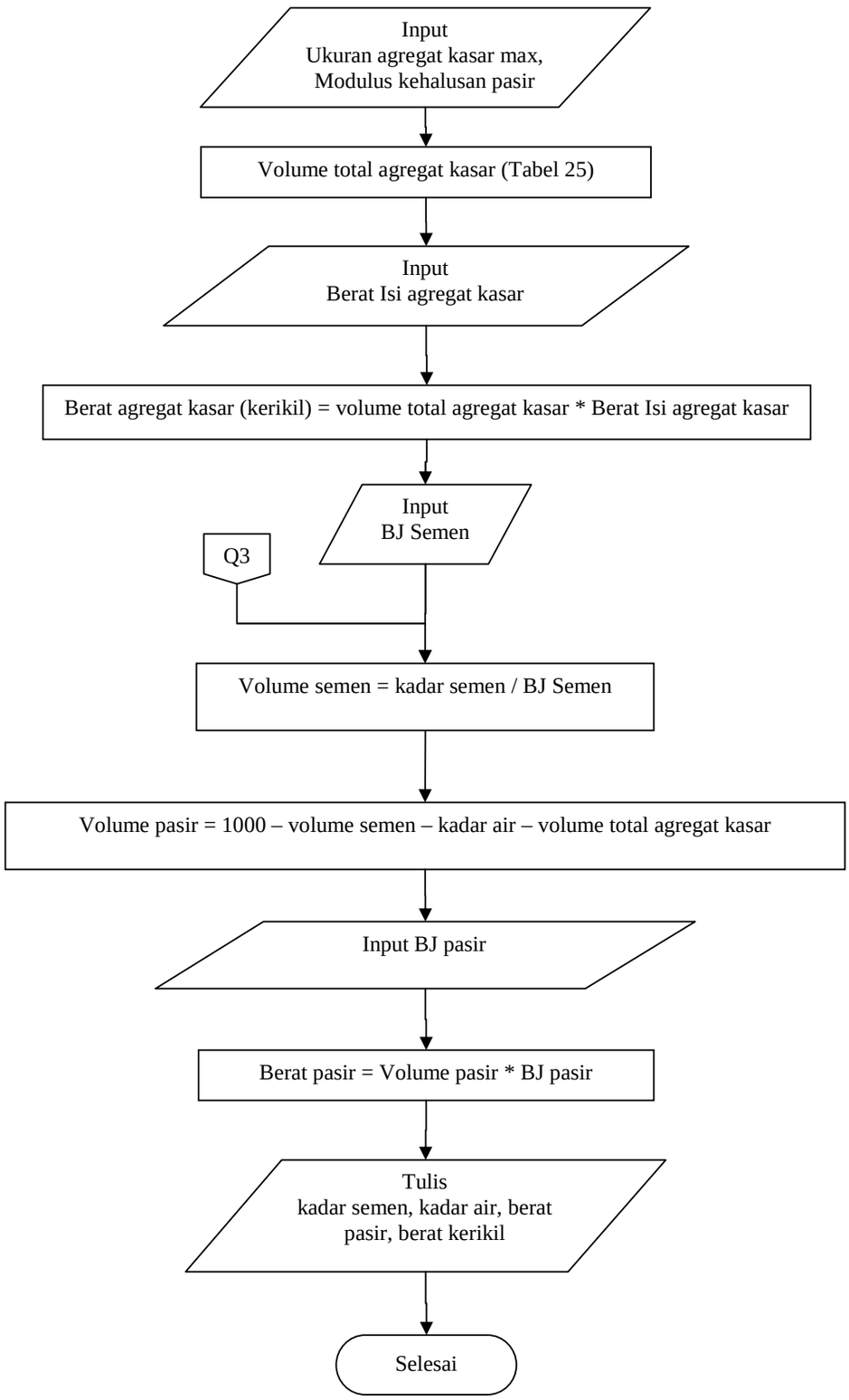
3.2.2 ALGORITMA METODE ACI



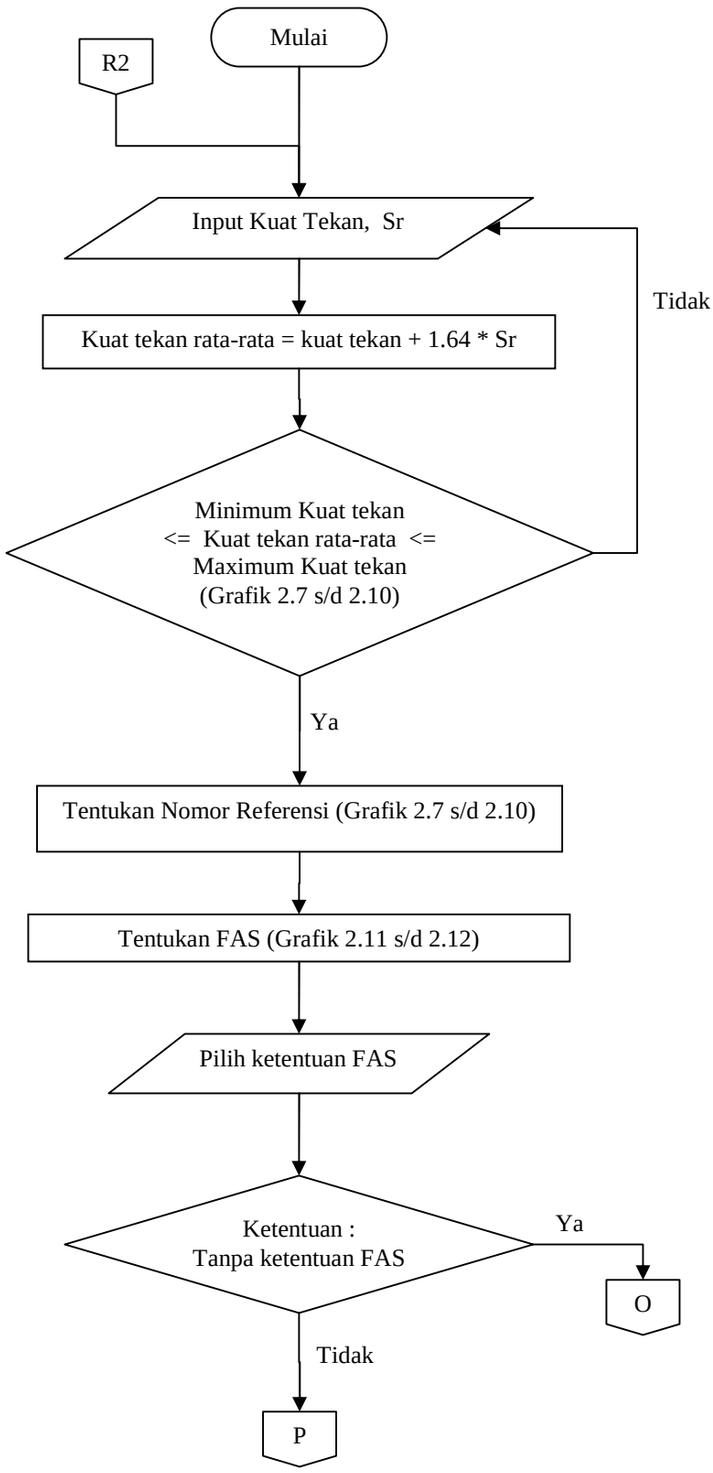

J
FAS
>=
FAS max
Ya
FAS = FAS max
I
Tidak
Input Slump, Ukuran Agregat Max
Tentukan Kadar Air (Tabel 2.3)
Kadar Semen = Kadar Air Bebas / FAS
Pilih ketentuan kadar semen
K

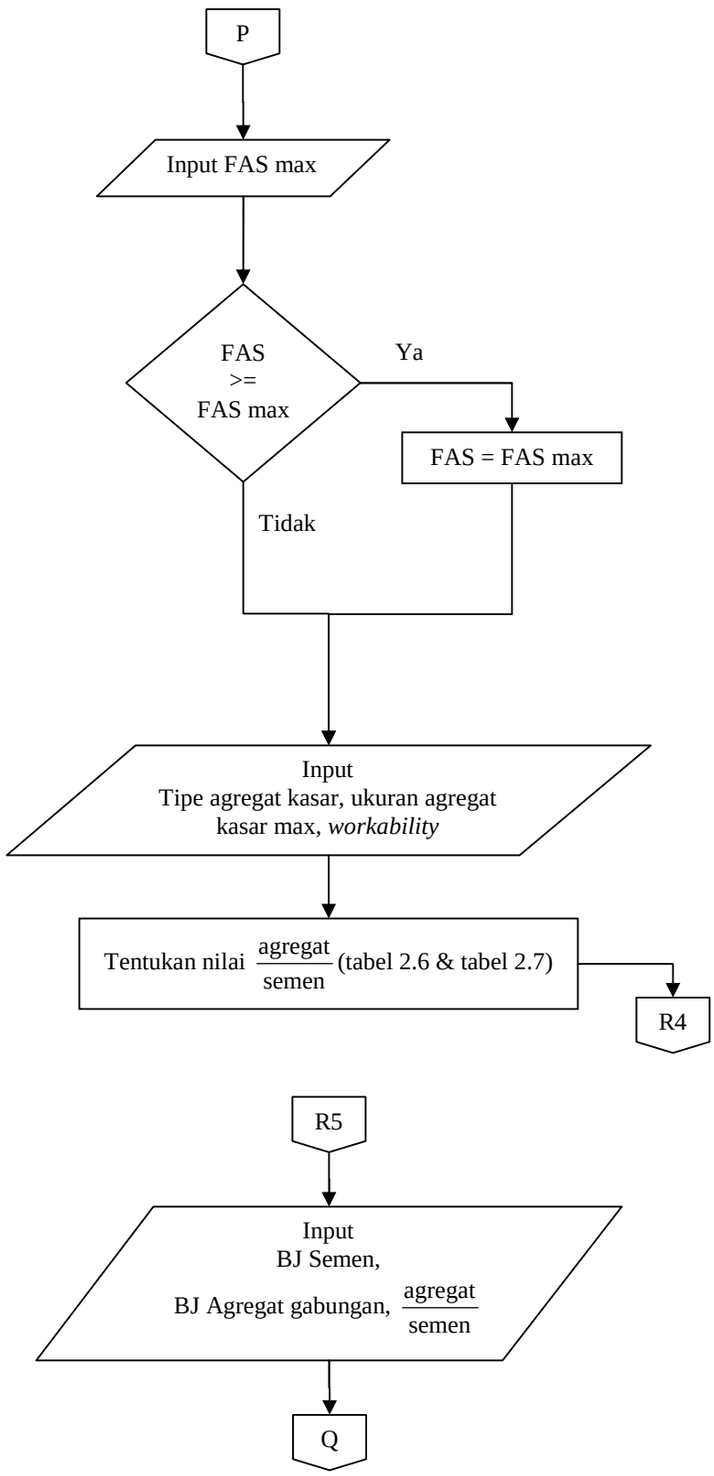

K
Ketentuan :
Tanpa ketentuan
kadar semen
Ya
L
Tidak
Ketentuan :
Kadar semen max
Ya
Input :
Kadar semen max
Tidak
Kadar semen
>=
kadar semen max
Ya
Kadar semen
=
Kadar semen max
Tidak
M
Ketentuan :
Kadar semen min
Ya
Input
Kadar semen min
Tidak
Kadar semen
<=
kadar semen min
Ya
Kadar semen
=
Kadar semen min
Tidak
Q1
N

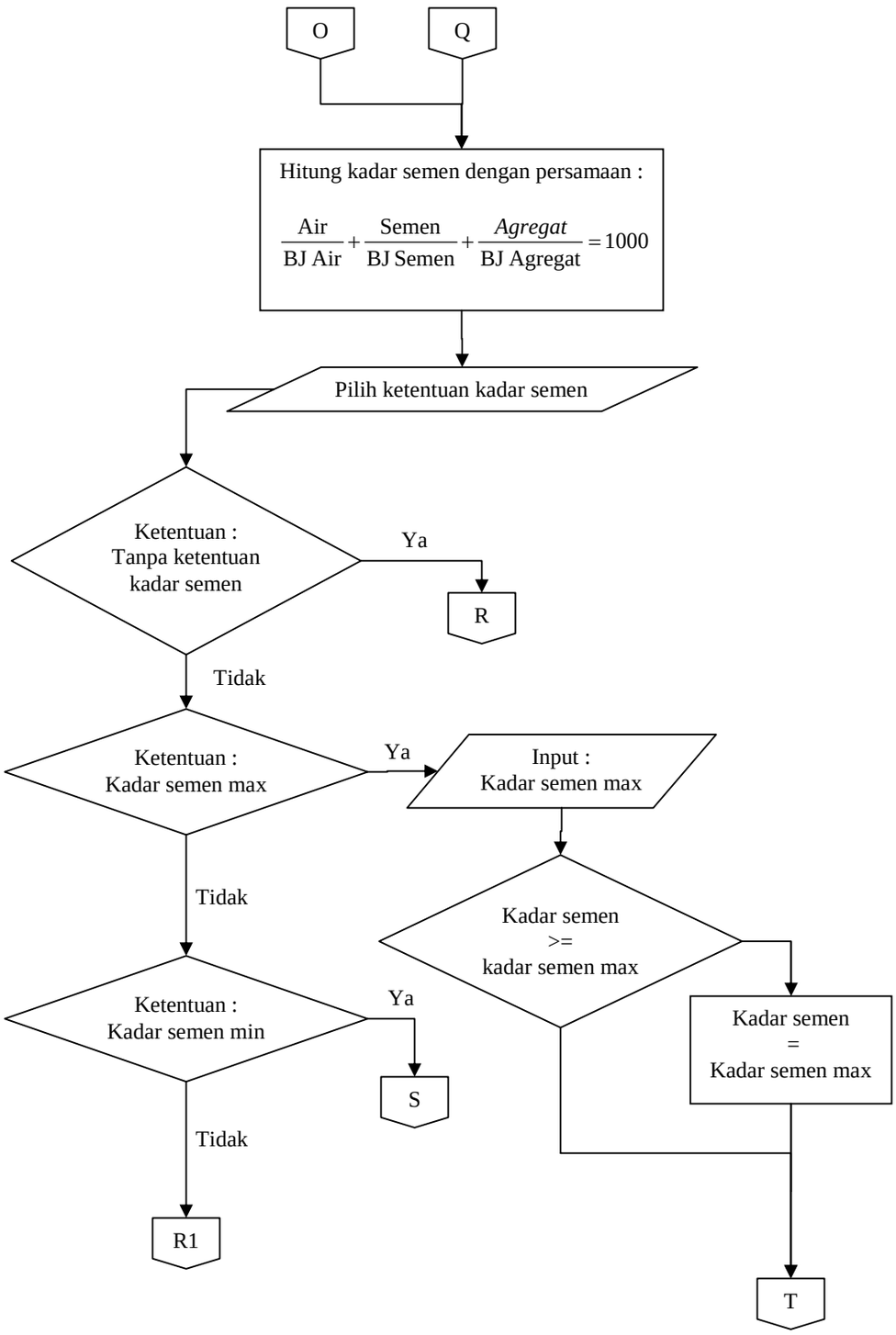

Q1
Input
kadar semen max,
kadar semen min
kadar semen min
<= Kadar semen <=
kadar semen max
Tidak
Q2
Ya
L
M
N
Q3

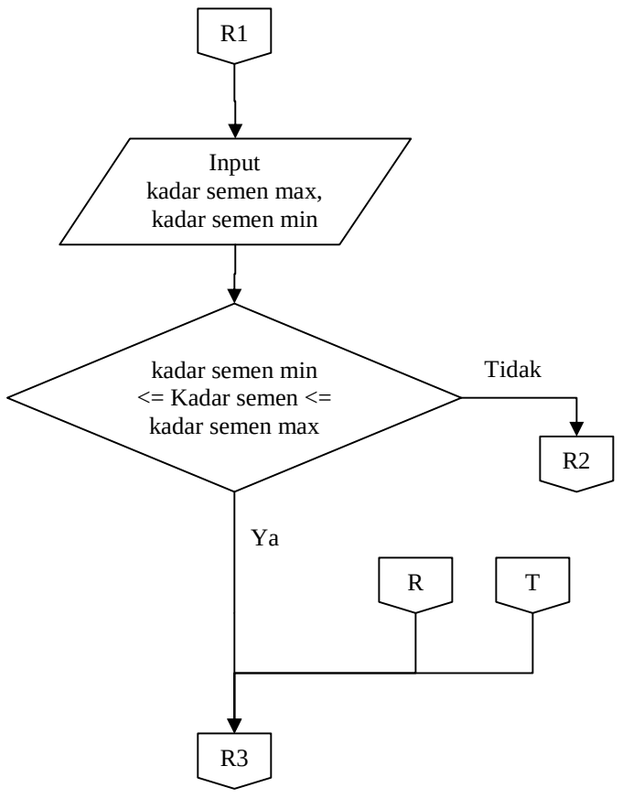

Input
Ukuran agregat kasar max,
Modulus kehalusan pasir
Volume total agregat kasar (Tabel 25)
Input
Berat Isi agregat kasar
Berat agregat kasar (kerikil) = volume total agregat kasar * Berat Isi agregat kasar
Input
BJ Semen
Q3
Volume semen = kadar semen / BJ Semen
Volume pasir = 1000 – volume semen – kadar air – volume total agregat kasar
Input BJ pasir
Berat pasir = Volume pasir * BJ pasir
Tulis
kadar semen, kadar air, berat
pasir, berat kerikil
Selesai

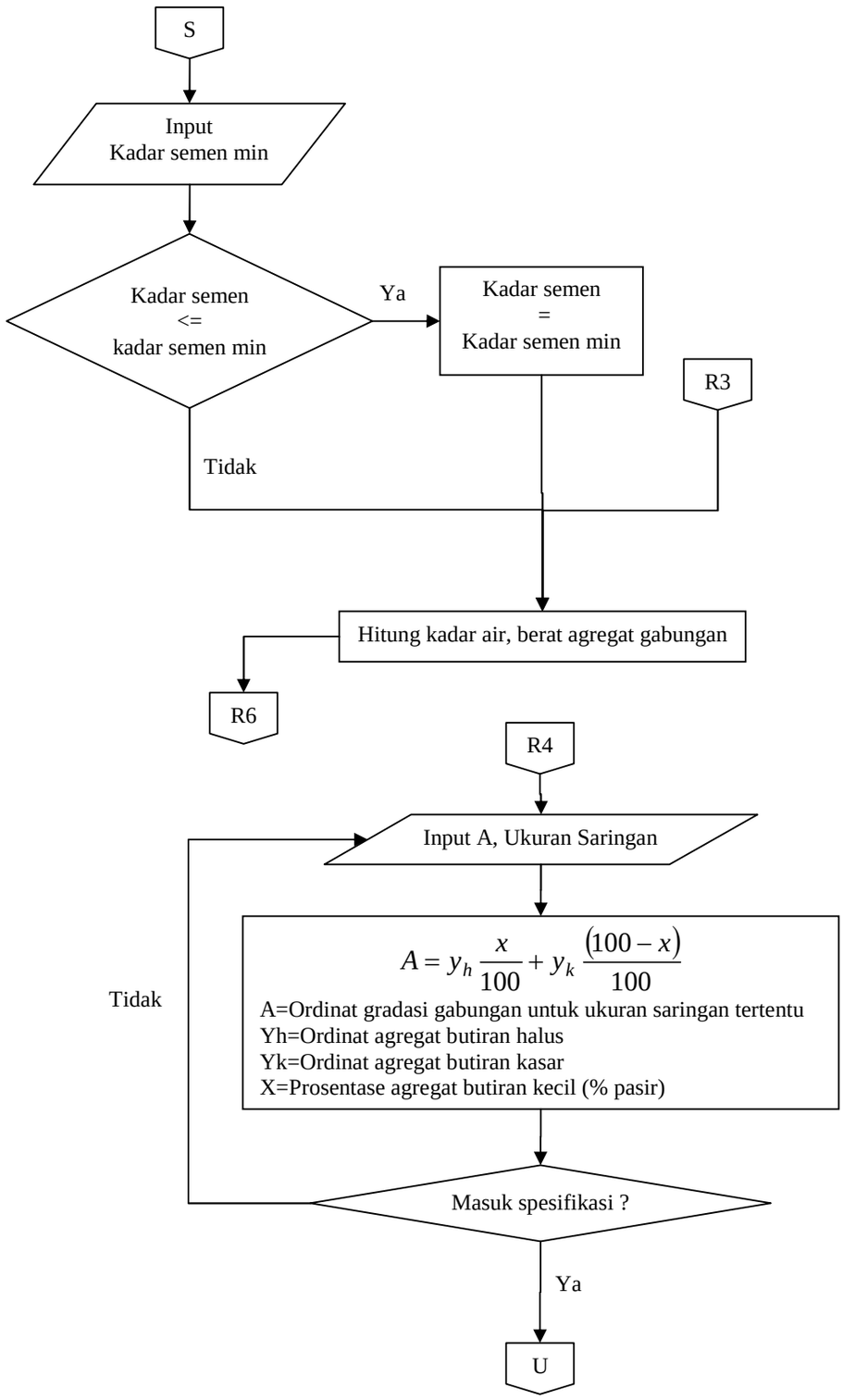
3.2.3 ALGORITMA METODE SHACKLOCK

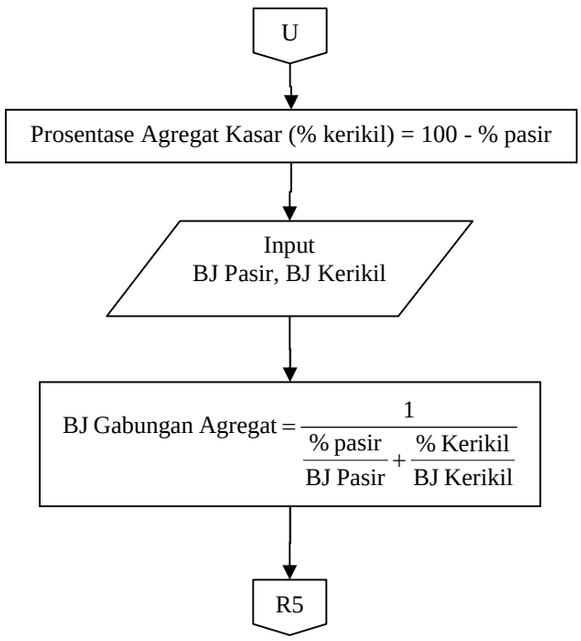


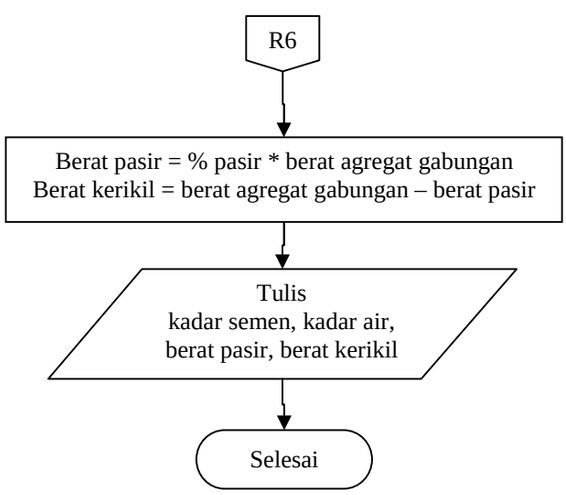

P
Input FAS max
FAS
>=
FAS max
Ya
FAS = FAS max
Tidak
Input
Tipe agregat kasar, ukuran agregat kasar max, *workability*
Tentukan nilai agregat/semen (tabel 2.6 & tabel 2.7)
R4
R5
Input
BJ Semen,
BJ Agregat gabungan, agregat/semen
Q


O
Q
Hitung kadar semen dengan persamaan :
Air / BJ Air + Semen / BJ Semen + Agregat / BJ Agregat = 1000
Pilih ketentuan kadar semen
Ketentuan :
Tanpa ketentuan
kadar semen
Ya
R
Tidak
Ketentuan :
Kadar semen max
Ya
Input :
Kadar semen max
Tidak
Kadar semen
>=
kadar semen max
Kadar semen
=
Kadar semen max
Ketentuan :
Kadar semen min
Ya
S
Tidak
R1
T


R1
Input
kadar semen max,
kadar semen min
kadar semen min
<= Kadar semen <=
kadar semen max
Tidak
R2
Ya
R
T
R3


S
Input
Kadar semen min
Kadar semen
<=
kadar semen min
Ya
Kadar semen
=
Kadar semen min
R3
Tidak
Hitung kadar air, berat agregat gabungan
R6
R4
Input A, Ukuran Saringan
$A = y_h \frac{x}{100} + y_k \frac{(100 - x)}{100}$
A=Ordinat gradasi gabungan untuk ukuran saringan tertentu
Yh=Ordinat agregat butiran halus
Yk=Ordinat agregat butiran kasar
X=Prosentase agregat butiran kecil (% pasir)
Tidak
Masuk spesifikasi ?
Ya
U


U
Prosentase Agregat Kasar (% kerikil) = 100 - % pasir
Input
BJ Pasir, BJ Kerikil
BJ Gabungan Agregat = 1 / (% pasir / BJ Pasir + % Kerikil / BJ Kerikil)
R5


R6
Berat pasir = % pasir * berat agregat gabungan
Berat kerikil = berat agregat gabungan – berat pasir
Tulis
kadar semen, kadar air,
berat pasir, berat kerikil
Selesai