

repository.ub.ac

IMPLEMENTASI ALGORITMA PENCARIAN AKAR KUADRAT BILANGAN POSITIF

SKRIPSI

KONSENTRASI REKAYASA KOMPUTER

Diajukan Untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Teknik



Disusun oleh:

MUHAMMAD IQBAL W.

NIM. 0510633057

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2012

repository.ub.ac

IMPLEMENTASI ALGORITMA PENCARIAN AKAR KUADRAT BILANGAN POSITIF

SKRIPSI

KONSENTRASI REKAYASA KOMPUTER

*Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik*



Disusun oleh :

MUHAMMAD IQBAL W.

NIM. 0510633057

Telah diperiksa dan disetujui oleh :

Pembimbing I

Pembimbing II

Waru Djuriatno, ST., MT.

NIP. 19690725 199702 1 001

Muhammad Aswin, Ir.,MT.

NIP. 19640626 1990021 1 001

LEMBAR PENGESAHAN
IMPLEMENTASI ALGORITMA PENCARIAN AKAR
KUADRAT BILANGAN POSITIF

SKRIPSI

KONSENTRASI REKAYASA KOMPUTER

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh :

MUHAMMAD IQBAL W.

NIM. 0510633057

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 10 Agustus 2012

Penguji I

Penguji II

Mochammad Rif'an, S.T., M.T

Adharul Muttaqin, S.T., M.T

NIP. 19710301 200012 1 001

NIP. 19760121 200501 1 001

Penguji III

A. Abdul Razak, S.T., M.T., M. Eng

NIP. -

Mengetahui,

Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono, MS

NIP. 19580728 198701 1 001

PENGANTAR

Puji syukur ke hadirat Allah SWT, atas nikmat, hidayah serta kasih sayangnya sehingga penulis dapat menyelesaikan skripsi dengan judul "*Implementasi Algoritma Pencarian Akar Kuadrat Bilangan Positif*". Hanya kepada-Nya kita menyembah dan memohon. Serta sholawat terhadap junjungan Rasulullah Muhammad SAW, keluarga, sahabat serta seluruh ummatnya. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Konsentrasi Teknik Rekayasa Komputer Fakultas Teknik Universitas Brawijaya Malang. Tidak banyak yang bisa penulis sampaikan kecuali ungkapan terima kasih kepada berbagai pihak yang telah memberikan bimbingan, arahan, dan dukungan hingga penulisan skripsi ini dapat terselesaikan. Pada kesempatan kali ini, penulis mengucapkan banyak terima kasih kepada:

1. Kedua orang tua penulis, papa dan mama, yang selalu tidak lepas dari doa dan harapan untuk terselesaikannya skripsi ini dan terus memberikan dorongan moral dan kasih sayangnya tiada akhir. Tak lupa juga penulis berterima kasih untuk segenap keluarga besar penulis.
2. Bapak Dr. Ir. Sholeh Hadi Pramono, MS. dan Bapak M. Azis Muslim, ST., MT., Ph.D. selaku Ketua dan Sekretaris Jurusan Teknik Elektro.
3. Bapak Waru Djuriatno, ST., MT., selaku Dosen Pembimbing I yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan skripsi ini.
4. Bapak M. Aswin, Ir., MT selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan skripsi ini.
5. Bapak Waru Djuriatno, ST., MT. selaku KKDK Teknik Informatika dan Komputer serta segenap bapak dan ibu dosen di jurusan elektro, serta staf administrasi.
6. Bapak M. Rif'an S.T., M.T., selaku Ketua Program Studi Jurusan Teknik Elektro yang telah banyak memotivasi saya dalam penyusunan skripsi ini.
7. Seluruh rekan yang banyak membantu skripsi saya, Dena, Fahmanda, Jefry Wijaya, Sanata, Muhzar, Irhamni, Ade Tri, Ahmad Afriyanto, Bagus, dan Ferdian.
8. Teman-teman angkatan 05 yang selalu menemani, memberikan doa, semangat agar skripsi ini cepat selesai.

9. Teman-teman jurusan elektro dari angkatan tua dan muda.
10. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan dan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Amin.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penulis maupun pihak lain yang menggunakannya.

Malang, 8 Agustus 2012

Penulis



DAFTAR ISI

PENGANTAR..... i

DAFTAR ISI..... iii

DAFTAR TABEL..... v

DAFTAR GAMBAR.....vi

ABSTRAK..... vii

BAB I PENDAHULUAN..... 1

 1.1 Latar Belakang..... 1

 1.2 Rumusan Masalah..... 2

 1.3 Batasan Masalah..... 2

 1.4 Tujuan..... 2

 1.5 Manfaat..... 3

 1.6 Sistematika Penulisan..... 3

BAB II TINJAUAN PUSTAKA..... 5

 2.1 Metode Numerik..... 5

 2.2 Akar Kuadrat..... 6

 2.3 Galat (Error).....7

 2.4 Metode Babylonian..... 8

BAB III Metodologi Penelitian..... 9

 3.1 Studi Literatur..... 9

 3.2 Perancangan..... 9

 3.3 Implementasi..... 12

 3.4 Pengujian dan Analisis..... 12

 3.5 Pengambilan Kesimpulan dan Saran..... 12

BAB IV PERANCANGAN13

 4.1 Perancangan sistem13

 4.1.1 Diagram Konteks13

 4.2 Cara Kerja Sistem14

 4.2.1 Input Data16

 4.2.2 Mengolah Data Dengan Metode Babylonian..... 16

 4.2.3 Menentukan Error17

 4.2.4 Output Data18

BAB V IMPLEMENTASI dan PENGUJIAN..... 19

 5.1 Lingkungan Implementasi 19

 5.2 Algoritma Sistem 19

 5.2.1 Implementasi Metode Babylonian..... 19

 5.2.2 Implementasi Algoritma pada Bahasa Pemrograman..... 21

 5.3 Pengujian..... 25

BAB VI KESIMPULAN DAN SARAN 32

 7.1 Kesimpulan 32

7.2 Saran..... 32
DAFTAR PUSTAKA.....33
LAMPIRAN.....34



DAFTAR TABEL

Tabel 5.1 Hasil Uji Program (bulat).....29
Tabel 5.2 Hasil Uji Program ke-2..... 30
Tabel 5.3 Hasil Uji Program ke-3..... 31



DAFTAR GAMBAR

Gambar Diagram Alur Algoritma.....	11
Gambar 4.1 Diagram Konteks.....	13
Gambar 4.2 Diagram Alir Sistem Secara Umum.....	15
Gambar 4.3 Diagram Alir Metode Babylonian.....	16
Gambar 4.4 Diagram Alir Menentukan Error.....	17
Gambar 4.5 Diagram Alir Output.....	18
Gambar 5.1 Antarmuka Program.....	25
Gambar 5.2 Hasil Program.....	25
Gambar 5.3 Program dengan Input Data 2.....	26
Gambar 5.4 Program dengan Input Data 43.8.....	26
Gambar 5.5 Program dengan Input Data 17.1.....	27
Gambar 5.6 Program dengan Input Data 5.987.....	27
Gambar 5.7 Program dengan Input Data 197.3.....	28
Gambar 5.8 Program dengan Input Data 142.....	28



ABSTRAK

Muhammad Iqbal W, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juli 2012, Implementasi Algoritma Pencarian Akar Kuadrat Bilangan Positif, Dosen pembimbing: Waru Djuriatno, ST., MT., Muhammad Aswin, Ir., MT.,

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Algoritma tidak hanya dapat diterapkan dalam dunia komputer. Tugas akhir ini membahas mengenai algoritma yang digunakan untuk mencari akar kuadrat bilangan positif dengan menggunakan metode numerik yang telah ada yaitu metode babylonian. Perangkat lunak ini dibangun pada perangkat desktop dengan menggunakan bahasa pemrograman C . Sistem ini bekerja dengan cara menentukan nilai akar kuadrat yang akan dicari kemudian mengolah nilai tersebut menggunakan algoritma babylonian. Berdasarkan hasil pengujian dan analisis sistem, perangkat lunak ini berhasil menjalankan semua fungsinya dengan benar, memberikan nilai yang mendekati nilai sebenarnya dengan kualitas error tertentu.

Kata kunci: Algoritma, Akar, Kuadrat.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Matematika yang merupakan salah satu alat dalam menyelesaikan suatu fenomena, biasanya berupa suatu rumusan. Artinya suatu fenomena dapat diselesaikan apabila sudah dirumuskan ke dalam bahasa matematika. Pada umumnya persoalan yang melibatkan model matematika banyak muncul dalam berbagai disiplin ilmu pengetahuan, seperti bidang fisika, kimia, ekonomi, atau pada persoalan rekayasa (*engineering*) seperti teknik sipil, teknik mesin, teknik elektro, dan sebagainya. Seringkali model matematika yang rumit ini tidak dapat diselesaikan dengan metode analitik yang sudah umum untuk mendapatkan solusi sejatinya (solusi eksak).

Metode numerik adalah teknik untuk menyelesaikan permasalahan-permasalahan yang diformulasikan secara matematis dengan cara operasi hitungan/aritmetika biasa (tambah, kurang, bagi, dan kali). Berbagai permasalahan dalam bidang ilmu pengetahuan dan teknologi dapat digambarkan dalam bentuk persamaan matematika. Apabila persamaan tersebut mempunyai bentuk sederhana, penyelesaiannya dapat dilakukan secara analitis. Tetapi pada umumnya bentuk persamaan sulit diselesaikan secara analitis, sehingga penyelesaiannya dilakukan secara numerik.

Algoritma adalah sekumpulan aturan-aturan berhingga yang memberikan sederetan operasi-operasi untuk menyelesaikan suatu jenis masalah yang khusus. Menurut Rinaldi Munir, algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Berdasarkan dua pengertian algoritma di atas, dapat disimpulkan bahwa algoritma merupakan suatu istilah yang luas, yang tidak hanya berkaitan dengan dunia komputer.

Salah satu penerapan metode numerik dalam perhitungan aritmetika adalah mencari akar-akar kuadrat. Sejauh ini akar kuadrat (pada bilangan positif) sangat banyak ditemukan dan mudah diselesaikan tetapi hanya diselesaikan secara analitik.

Dari beberapa metode tersebut pada dasarnya mempunyai tujuan yang sama, yaitu menentukan akar-akar kuadrat dengan *error* yang paling kecil. Semua persoalan tersebut dapat diketahui dengan jelas, apabila sudah dibuktikan atau diterapkan dalam suatu contoh dan nantinya juga dibutuhkan pemikiran yang lebih mendalam lagi dalam penganalisisannya.

Dalam skripsi ini nantinya akan menggunakan metode algoritma estimasi akar kuadrat. Dengan menggunakan metode ini maka akan mampu menghasilkan hasil dan error lebih baik

1.2 Rumusan Masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang maka rumusan masalah dikhususkan pada :

- 1) Merancang algoritma estimasi akar kuadrat.
- 2) Merancang aplikasi untuk mencari akar kuadrat bilangan positif.
- 3) Bagaimana analisis hasil pencarian akar kuadrat pada bilangan positif dengan algoritma Estimasi akar kuadrat dengan menggunakan bahasa c.

1.3 Batasan Masalah

Batasan-batasan yang diberikan dalam penulisan skripsi ini adalah :

- 1) Bahasa pemrograman yang digunakan adalah bahasa pemrograman C.
- 2) Bilangan yang dicari adalah bilangan positif.

1.4 Tujuan

Tujuan penelitian adalah untuk merancang dan membuat aplikasi algoritma pencarian akar kuadrat pada bilangan positif dengan hasil yang baik.

1.5 Manfaat

Manfaat yang diharapkan dapat diperoleh melalui pengerjaan skripsi ini adalah

1. Bagi Penulis
 - a. Menerapkan ilmu yang telah diperoleh dalam Teknik Elektro konsentrasi sistem informatika dan komputer Universitas Brawijaya.
 - b. Mengembangkan sebuah aplikasi sistem yang tepat sehingga mendapatkan hasil yang maksimal dalam menerapkan keilmuan di bidang teknik.
2. Bagi Pengguna

Sebagai tambahan pengetahuan bidang matematika, khususnya metode numerik mengenai cara pencarian akar kuadrat menggunakan algoritma estimasi nilai akar kuadrat dengan menggunakan bahasa C.

1.6 Sistematika Penulisan

Sistematika penulisan laporan skripsi ini adalah sebagai berikut :

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika pembahasan.

BAB II Dasar Teori

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan aplikasi.

BAB III Metodologi

Berisi tentang metode penelitian yang digunakan dalam perancangan dan pengujian sistem.

BAB IV Analisis Kebutuhan dan Perancangan

Membahas tentang analisa kebutuhan dari sistem dan kemudian merancang hal-hal yang berhubungan dengan analisa tersebut.

BAB V Implementasi dan Pengujian

Bagian ini berisi penjelasan tentang implementasi yang telah dilakukan dan batasan-batasan implementasi dan tentang pembahasan mengenai pengujian perancangan algoritma dan pengujian aplikasi pencarian akar kuadrat menggunakan algoritma estimasi nilai akar kuadrat.

BAB VI Penutup

Bagian ini berisi kesimpulan dan saran. Kesimpulan didasarkan atas pengujian dan analisis yang dilakukan di dalam proses penelitian.



BAB II

TINJAUAN PUSTAKA

2.1 Metode Numerik

Metode numerik merupakan salah satu bidang ilmu pengetahuan, khususnya matematika rekayasa, yang menggunakan bilangan untuk menirukan proses matematika, yang selanjutnya telah dirumuskan untuk menirukan keadaan sebenarnya. Di dalam kegiatan rekayasa dan penelitian, setiap analisis diharapkan dapat menghasilkan bilangan, yang diperlukan dalam perencanaan teknik ataupun penghayatan masalah (Djojodihardjo, 2000: 1)

Selain persoalan rekayasa (*engineering*) seperti teknik sipil, teknik mesin, teknik elektro ternyata juga ada disiplin ilmu lainnya yakni bidang fisika, kimia, ekonomi, dan lain-lain yang semuanya juga melibatkan model matematika dalam pemecahannya. Sering kali model matematika tersebut muncul dalam bentuk yang tidak ideal atau rumit. Model matematika yang rumit ini ada kalanya tidak dapat diselesaikan dengan metode analitik yang sudah umum untuk mendapatkan solusi sejatinya (*exact solution*). Yang dimaksud dengan metode analitik adalah metode penyelesaian model matematika dengan rumus-rumus aljabar yang sudah ada.

Beberapa persoalan yang melibatkan model matematika dalam pemecahannya antara lain menentukan akar-akar kuadrat, menghitung nilai suatu fungsi bila rumus fungsinya sendiri tidak diketahui, untuk masalah pengamatan fenomena alam, baik berupa eksperimen di laboratorium maupun penelitian di lapangan yang melibatkan beberapa parameter (misalnya suhu, tekanan, dan sebagainya). Pengamat tidak mengetahui relasi yang menghubungkan parameter-parameter itu. Pengamat hanya dapat mengukur nilai-nilai parameter tersebut dengan menggunakan alat ukur seperti sensor, termometer, barometer, dan sebagainya. Tidak ada satupun metode analitik yang tersedia untuk menyelesaikan persoalan jenis ini.

Dengan melihat persoalan-persoalan di atas, jika metode analitik tidak dapat lagi digunakan, maka persoalan sebenarnya masih dapat dicari dengan menggunakan metode numerik. Metode numerik adalah teknik yang digunakan untuk

memformulasikan persoalan matematika sehingga dapat dipecahkan dengan operasi perhitungan/ aritmetika biasa (tambah, kurang, kali, dan bagi).

Metode artinya cara, sedangkan numerik artinya angka. Jadi metode numerik secara harfiah berarti cara berhitung dengan menggunakan angka-angka (Munir, 2006: 1-5). Berdasarkan definisi harfiah tersebut, dapat diketahui bahwa metode numerik yang berarti cara berhitung dengan menggunakan angka-angka ini sangat erat hubungannya dengan matematika. Karena matematika itu pada dasarnya adalah disiplin ilmu yang berkaitan dengan pekerjaan menghitung, sehingga tidak salah jika kemudian ada yang menyebut matematika adalah ilmu hitung atau *ilmu hisab*. Dalam pekerjaan menghitung, selain dibutuhkan bilangan, di dalamnya juga terdapat operasi hitung/ aritmetika biasa (tambah, kurang, kali, dan bagi).

2.2 Akar kuadrat

Di dalam matematika, akar kuadrat dari bilangan x sama dengan bilangan r sedemikian sehingga $r^2 = x$, atau, di dalam perkataan lain, bilangan r yang bila dikuadratkan (hasil kali dengan bilangan itu sendiri) sama dengan x .

Setiap bilangan real tak-negatif, katakanlah x memiliki akar kuadrat tak-negatif yang tunggal, disebut akar kuadrat utama, yang dilambangkan oleh akar ke- n sebagai $\sqrt[n]{x}$. Akar kuadrat dapat juga dituliskan dengan notasi eksponen, sebagai $x^{1/2}$. Misalnya, akar kuadrat utama dari 9 adalah 3, dituliskan dengan $\sqrt{9}=3$, karena $3^2 = 3 \times 3 = 9$ dan 3 tak-negatif. Bagaimanapun, akar kuadrat utama dari sebuah bilangan positif hanya satu dari dua akar kuadratnya.

Setiap bilangan positif x memiliki dua akar kuadrat. Salah satunya adalah \sqrt{x} , yakni yang bernilai positif, sementara yang lainnya adalah $-\sqrt{x}$, yakni yang bernilai negatif. Kedua-dua akar kuadrat itu dilambangkan dengan $\pm\sqrt{x}$. Akar kuadrat dari bilangan negatif dibahas di dalam kerangka kajian bilangan kompleks. Lebih umum lagi, akar kuadrat dapat dipandang dari beraneka konteks di mana notasi "penguadratan" beberapa objek matematika didefinisi (termasuk aljabar matriks, dll).

Akar kuadrat dari bilangan bulat yang bukan merupakan kuadrat sempurna adalah selalu *bilangan irasional* (disebut juga bilangan takrasional: bilangan yang

tidak dapat dinyatakan sebagai hasil bagi dari dua bilangan bulat. Misalnya, $\sqrt{2}$ tidak dapat dituliskan secara tepat oleh m/n , di mana n dan m adalah bilangan bulat. Meskipun demikian, ia adalah nilai yang pasti dari panjang diagonal sebuah persegi yang panjang sisinya sama dengan 1.

2.3 Galat (error)

Ada tiga sumber utama galat dalam suatu perhitungan numerik, yaitu: galat bawaan (inheren), galat pemotongan, dan galat pembulatan.

a. Galat Inheren (bawaan)

Galat inheren adalah galat dalam nilai data, disebabkan oleh ketidakpastian dalam pengukuran, kekeliruan atau oleh perlunya pendekatan untuk menyatakan suatu bilangan yang angkanya tidak secara tepat dapat dinyatakan dengan banyaknya angka yang tersedia. Galat ini biasanya berhubungan dengan galat pada data yang dioperasikan oleh komputer dengan beberapa prosedur numerik.

b. Galat Pemotongan

Galat pemotongan adalah galat yang terjadi karena hanya diperhitungkannya beberapa suku pertama. Galat yang terjadi karena tidak dilakukannya hitungan sesuai dengan prosedur matematika yang benar. Sebagai contoh suatu proses tak terhingga diganti dengan proses berhingga.

c. Galat Pembulatan

Galat yang terjadi karena tidak diperhitungkannya, beberapa angka terakhir dari suatu bilangan. Kesalahan ini terjadi apabila bilangan perkiraan digunakan untuk menggantikan bilangan eksak. Suatu bilangan dibulatkan pada posisi tersebut nol. Sedang angka pada posisi ke n tersebut tidak berubah atau dinaikkan satu digit yang tergantung apakah nilai tersebut lebih kecil atau lebih besar dari setengah dari angka posisi ke n .

2.4 Babylonian Method

Metode ini merupakan metode pertama digunakan untuk mendekati \sqrt{S} , yang dikenal sebagai "metode Babel", atau "metode Heron", metode ini dikenal (berabad-abad mendahului) sebelum metode yang dibuat oleh Newton. Menurut metode ini, dasarnya bahwa jika x lebih besar dari akar kuadrat S maka S/x akan menjadi lebih kecil sehingga rata-rata dari dua angka tersebut diharapkan dapat memberikan pendekatan yang lebih baik. Algoritma ini merupakan algoritma kuadratik konvergen, yang berarti bahwa jumlah digit yang benar dari pendekatan tersebut berlipat ganda dengan setiap iterasi, seperti berikut:

1. Mulailah dengan sebuah nilai awal acak positif (lebih dekat dengan akar kuadrat yang sebenarnya dari S , lebih baik).
2. Misalkan x_{n+1} adalah rata-rata dari x_n dan S/x_n .
3. Ulangi langkah 2 sampai akurasi yang diinginkan tercapai.

Hal ini juga dapat direpresentasikan sebagai:

$$x_0 \approx \sqrt{S}.$$

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{S}{x_n} \right), \sqrt{S} = \lim_{n \rightarrow \infty} x_n.$$

BAB III

Metodologi Penelitian

Bab ini menjelaskan mengenai langkah-langkah yang dilakukan untuk membuat aplikasi algoritma pencarian akar kuadrat pada bilangan positif. Langkah-langkah yang diperlukan antara lain studi literatur, perancangan sistem, penentuan batasan algoritma yang digunakan, implementasi, pengujian dan analisis, pengambilan kesimpulan dan saran, serta penulisan laporan.

3.1 Studi Literatur

Berupa kajian pustaka yang mendukung pembuatan aplikasi yang berupa :

1. Kajian pustaka mengenai metode numerik.
2. Kajian pustaka mengenai algoritma estimasi nilai akar kuadrat.
3. Kajian pustaka mengenai struktur dan bahasa pemrograman yang digunakan yaitu bahasa C.

3.2 Perancangan

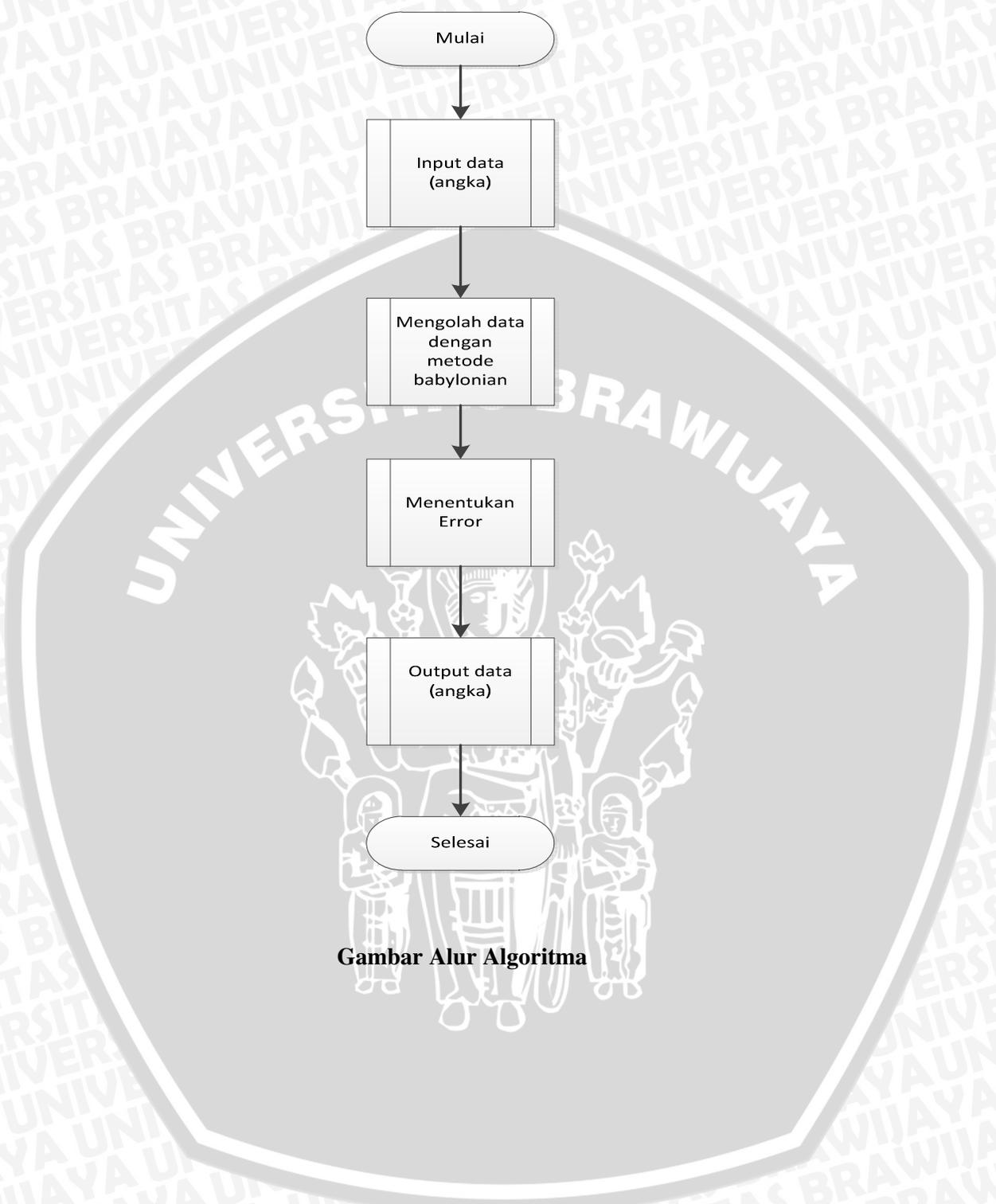
Secara garis besar perancangan terdiri dari perancangan sistem dan cara kerja sistem,. Perancangan sistem yaitu merancang blok sistem diagram untuk menggambarkan sistem secara global dan selanjutnya dilakukan analisis sesuai kebutuhan. Analisis kebutuhan digunakan untuk menentukan kebutuhan-kebutuhan fungsi, algoritma yang digunakan, dan batasan perancangan dan implementasi. Berikut ini adalah alur utama dari sistem :

Nilai $y = \sqrt{S}$ dapat diperoleh dengan langkah-langkah berikut:

1. Tentukan nilai kesalahan yang diperbolehkan, misalnya $error = 0,0000001$
2. Menentukan nilai N yang sudah diketahui kuadratnya yangmana $N^2 \approx S$.
3. Mengecek kuadrat nilai N terhadap S , jika nilai kuadrat $N < S$ maka N adalah nilai perkiraan awal.
4. Dengan demikian, nilai yang hendak kita hitung terletak di antara N hingga S .
5. Maka, $X_t = N$.
6. Berdasarkan algoritma Babylonian, $X_{t+1} = \frac{1}{2} \left(X_t + \frac{S}{X_t} \right)$.

7. Permasalahannya adalah apakah nilai perkiraan tersebut terlalu besar dibandingkan nilai yang sebenarnya, ataukah terlalu kecil?
8. Setelah diperoleh batas X_t dan batas X_{t+1} yang baru, ulangi langkah ke-6, hingga diperoleh $|X_t - X_{t+1}| \leq (error)^2$, yaitu bahwa selisih antara nilai perkiraan dengan nilai sebenarnya tidaklah lebih besar dari besarnya kesalahan yang diperbolehkan (dalam hal ini adalah ketelitian).
9. Nilai $|X_t - X_{t+1}| = P$, P inilah yang merupakan nilai perkiraan untuk y dengan nilai kesalahan maksimumnya adalah sebesar $error$. Dari teori di atas, buatlah algoritma untuk mengestimasi nilai $y = n$ dengan tingkat kesalahan sebesar $error$. Nilai n diinput dari user.
10. Analisis:
Input : nilai n (dalam hal ini, n haruslah tidak negatif!).
Output : nilai perkiraan n , yaitu P .





Gambar Alur Algoritma

3.3 Implementasi

Proses selanjutnya adalah implementasi yakni, proses transformasi hasil perancangan pencarian akar kuadrat bialangan positif yang telah dibuat ke dalam kode (*coding*) sesuai dengan sintaks dari bahasa pemrograman yang digunakan, yaitu menerapkan bahasa pemrograman C.

3.4 Pengujian dan Analisis

Pengujian aplikasi yaitu untuk mengetahui kesesuaian analisa kebutuhan yang dibuat dengan implementasi aplikasi. Analisis sistem dilakukan dengan membandingkan sistem aplikasi yang dengan teori yang ada sehingga didapatkan suatu kesimpulan.

3.5 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktek. Kesimpulan diambil untuk menjawab rumusan masalah yang telah ditetapkan sebelumnya. Tahap yang paling akhir adalah pengambilan saran terhadap penelitian yang selanjutnya, sehingga bisa menyempurnakan kekurangan yang ada dan mengembangkan pada tingkat pokok kajian yang lebih lanjut.

BAB IV

PERANCANGAN

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan sistem dan cara kerja sistem.

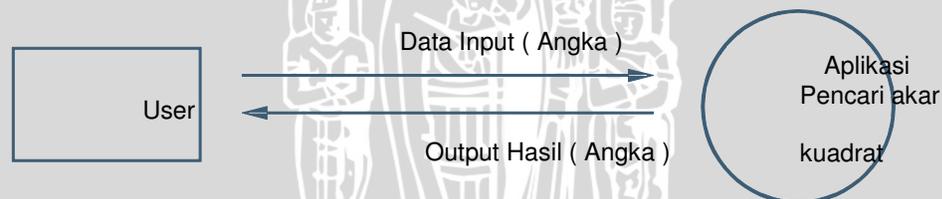
4.1 Perancangan Sistem

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui aplikasi sistem yang akan dibuat secara umum. Perancangan sistem ini akan didahului dengan pendefinisian pelaku atau user yang akan menggunakan program ini dan juga alat bantu yang digunakan, yaitu:

1. User : Pelaku yang akan memberikan input data dan juga menjalankan program dengan tujuan untuk mengetahui akar kuadrat bilangan positif.
2. Program : Perangkat lunak yang digunakan untuk mengolah data masukan menggunakan metode Babylonian dalam pencarian akar kuadrat.

4.1.1 Diagram Konteks

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan dengan entitas luar, masukan dan keluaran dari sistem. Diagram konteks aplikasi pencarian akar kuadrat bilangan positif ditunjukkan pada gambar 4.1.



Gambar 4.1 Diagram Konteks

4.2 Cara Kerja Sistem

Cara kerja aplikasi pencarian akar kuadrat ini adalah sebagai berikut:

Nilai $N = \sqrt{S}$ dapat diperoleh dengan langkah-langkah berikut:

11. Tentukan nilai S , ini dilakukan sebagai data masukan.
12. Mengolah data berdasarkan algoritma Babylonian, $X_{t+1} = \frac{1}{2} \left(X_t + \frac{S}{X_t} \right)$

13. Menentukan error, setelah diperoleh X_t dan X_{t+1} yang baru, hingga diperoleh $|X_t - X_{t+1}| \leq (\text{error})^2$, yaitu bahwa selisih antara nilai perkiraan dengan nilai sebenarnya tidaklah lebih besar dari besarnya kesalahan yang diperbolehkan (dalam hal ini adalah ketelitian).

14. Output data berupa nilai $|X_t - X_{t+1}| = P$, P inilah yang merupakan nilai perkiraan untuk y dengan nilai kesalahan maksimumnya adalah sebesar error . Dari teori di atas,

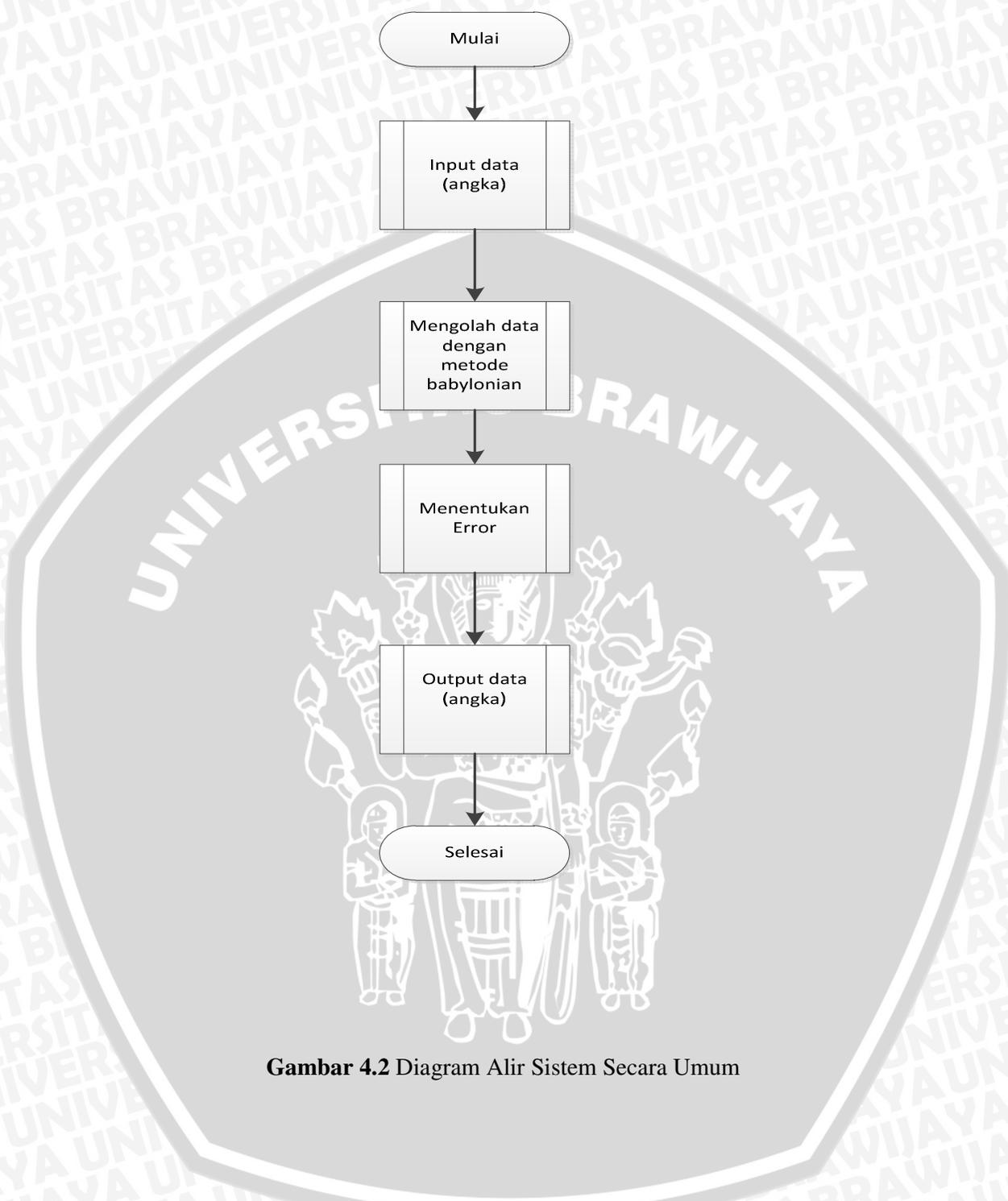
15. Analisis:

Input : nilai S . (dalam hal ini, S haruslah tidak negatif!).

Output : nilai perkiraan n , yaitu P .

UNIVERSITAS BRAWIJAYA





Gambar 4.2 Diagram Alir Sistem Secara Umum

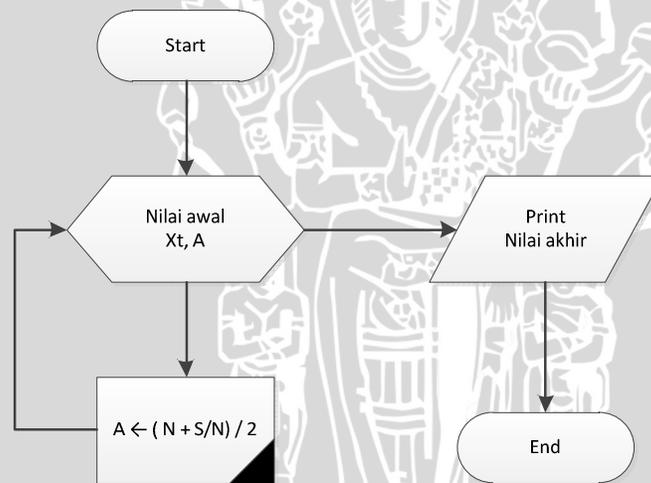
4.2.1 Input Data

Dalam Aplikasi ini, untuk menginput data, user langsung memasukkan atau memberikan nilai yang berupa angka. Dalam hal ini, angka atau bilangan yang digunakan adalah bilangan positif.

4.2.2 Mengolah data dengan Metode Babylonian

Ada beberapa metode untuk mendapatkan nilai akar kuadrat. Dalam aplikasi ini kita akan menggunakan metode babylonian. Dengan menggunakan metode ini kita bisa mendapatkan nilai awal perkiraan pencarian akar kuadrat. Adapun algoritma dalam metode Babylonian ini adalah sebagai berikut:

1. Maka, $X_t = N$, N adalah nilai awal perkiraan.
2. Berdasarkan algoritma Babylonian, $X_{t+1} = \frac{1}{2} \left(X_t + \frac{S}{X_t} \right)$.

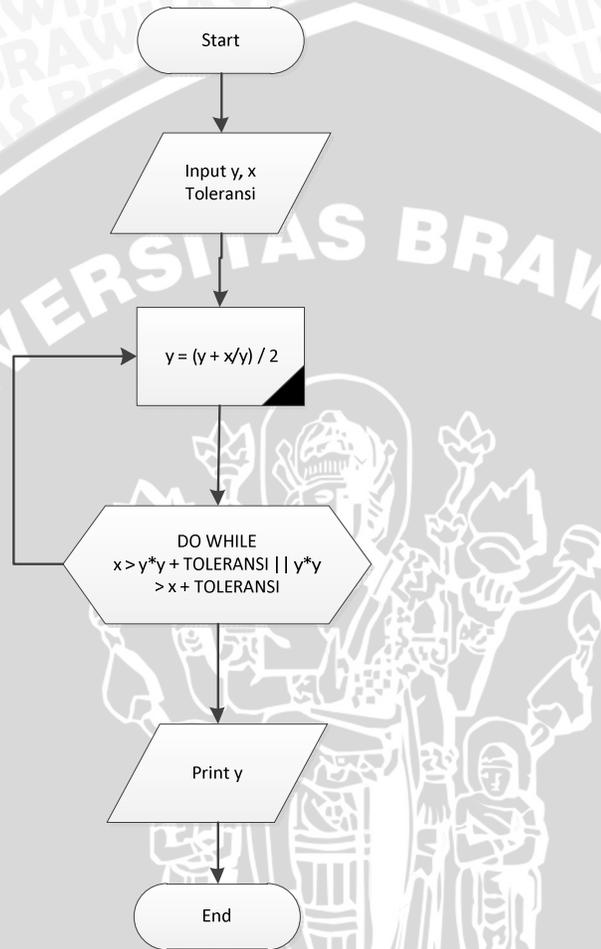


Gambar 4.3 Diagram Alir Metode Babylonian

Setelah mendapatkan nilai perhitungan dari metode Babylonian, maka selanjutnya nilai tersebut akan diproses untuk meminimalkan kesalahan dari nilai sebenarnya.

4.2.3 Menentukan Error

Dalam menentukan nilai error ini, aplikasi dibatasi dengan nilai yang telah ditentukan yaitu dengan membatasi bilangan atau angka dibelakang koma sebesar 10^{-15}

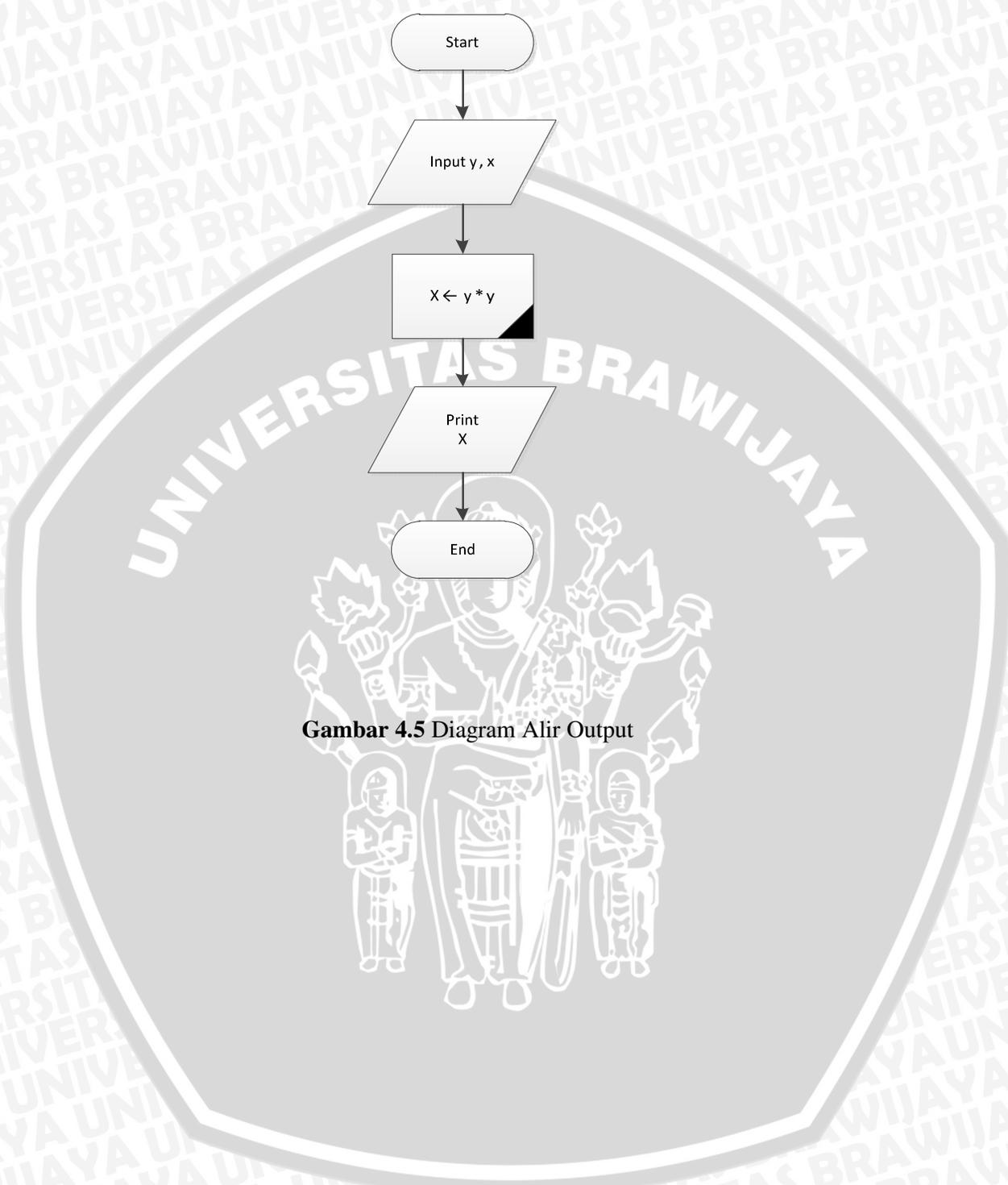


Gambar 4.4

Diagram Alir Menentukan Error

4.2.4 Output Data

Output data pada aplikasi ini yaitu dengan menampilkan hasil dari integrasi proses sebelumnya. Dalam hal ini menampilkan data tiap iterasi yang diperoleh hingga error yang telah ditentukan.



Gambar 4.5 Diagram Alir Output

BAB V

IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari perancangan perangkat lunak yang telah dibuat. Implementasi merupakan proses transformasi hasil perancangan perangkat lunak ke dalam kode (*coding*) sesuai dengan sintaks dari bahasa pemrograman yang digunakan. Pembahasan terdiri dari penjelasan tentang Lingkungan Implementasi, Implementasi Metode Babylonian, dan Implementasi Algoritma pada Bahasa Pemrograman.

Lingkungan Implementasi

Sistem dibuat dengan menggunakan aplikasi pemrograman Turbo C++ 4.5. Sistem diimplementasikan dengan menggunakan spesifikasi sebagai berikut:

5.1.1 Spesifikasi Perangkat Keras

- CPU : Intel M Processor (1.87 GHz)
- Memory : 2 GB RAM
- Hard Disk : 80 GB
- Graphics Adapter : ATI RADEON X300

5.1.2 Spesifikasi Perangkat Lunak

- Sistem operasi : Windows 7
- Aplikasi yang digunakan : Turbo C++ 4.5

5.2 Algoritma Sistem

Penyajian yang digunakan berupa urutan baris algoritma seperti kode pemrograman dan parameter yang digunakan dalam Tahap Implementasi ini berdasarkan perancangan sistem pada tahap Perancangan.

5.2.1 Implementasi Metode Babylonian

Sebelum data diolah, pada perancangan sistem algoritma yang digunakan adalah metode Babylonian. Menurut metode ini, dasarnya bahwa jika x lebih besar dari akar kuadrat S maka S/x akan menjadi lebih kecil

sehingga rata-rata dari dua angka tersebut diharapkan dapat memberikan pendekatan yang lebih baik. Ini merupakan algoritma kuadrat konvergen algoritma, yang berarti bahwa jumlah digit yang benar dari pendekatan tersebut berlipat ganda dengan setiap iterasi, seperti berikut:

4. Mulailah dengan sebuah nilai awal acak positif (lebih dekat dengan akar kuadrat yang sebenarnya dari S, lebih baik).
5. Misalkan x_{n+1} adalah rata-rata dari x_n dan S/x_n

Ulangi langkah 2 sampai akurasi yang diinginkan tercapai

$$x_0 \approx \sqrt{S}.$$

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{S}{x_n} \right),$$

$$\sqrt{S} = \lim_{n \rightarrow \infty} x_n.$$

Seperti pada contoh :

- Misal, _____
- Maka sebagai titik awal pencarian akar _____, _____

Sehingga, _____



$$x_4 = \frac{1}{2} \left(x_3 + \frac{S}{x_3} \right) = \frac{1}{2} \left(158.7786146 + \frac{2457}{158.7786146} \right) = 87.12649535$$

$$x_5 = \frac{1}{2} \left(x_4 + \frac{S}{x_4} \right) = \frac{1}{2} \left(87.12649535 + \frac{2457}{87.12649535} \right) = 57.66343609$$

$$x_6 = \frac{1}{2} \left(x_5 + \frac{S}{x_5} \right) = \frac{1}{2} \left(57.66343609 + \frac{2457}{57.66343609} \right) = 50.13246503$$

$$x_7 = \frac{1}{2} \left(x_6 + \frac{S}{x_6} \right) = \frac{1}{2} \left(50.13246503 + \frac{2457}{50.13246503} \right) = 49.57131172$$

$$x_8 = \frac{1}{2} \left(x_7 + \frac{S}{x_7} \right) = \frac{1}{2} \left(49.57131172 + \frac{2457}{49.57131172} \right) = 49.56813503$$

$$x_9 = \frac{1}{2} \left(x_8 + \frac{S}{x_8} \right) = \frac{1}{2} \left(49.56813503 + \frac{2457}{49.56813503} \right) = 49.56813493$$

$$x_{10} = \frac{1}{2} \left(x_9 + \frac{S}{x_9} \right) = \frac{1}{2} \left(49.56813493 + \frac{2457}{49.56813493} \right) = 49.56813493$$

Maka , $y = \sqrt{2457}$ adalah 49.56813493

5.2.2 Implementasi Algoritma pada Bahasa Pemrograman

Bahasa pemrograman dalam hal ini menggunakan bahasa pemrograman C. Pada aplikasi, algoritma yang digunakan sebelumnya akan diterapkan pada bahasa C, yaitu metode Babylonian. Berdasarkan pada perancangan, berikut adalah *source code* dari program :

1. Mendeklarasikan fungsi dari metode Babylonian.

```
#include <iostream.h>
#include <iomanip.h>

/*deklarasi fungsi babylon*/
double  babvlon(double):
```

2. Fungsi utama.

```
/*fungsi utama*/
int main()
{
    double X;
    cout << "Input bilangan yang akan dicari akarnya : ";
    cin >> X;
    cout << "=====\n";
    cout << "\t Y \t Y*Y \n";
    cout << "=====\n";
```

a. Panggil fungsi babylon dengan parameter nilai akar yang dicari.

```
/*panggil fungsi babylon */
```

b. Set nilai kepresisian (jika diperlukan).

```
/*set nilai kepresisian */
    cout << "Akar " << X << " adalah : " << setprecision(20)
<<y << endl;
```

3. Detail fungsi Babylonian.

```
/*detail fungsi babylon*/
double babylon(double x)
```

4. Batas nilai toleransi (epsilon), usahakan nilai jangan terlalu besar atau terlalu kecil.

```

/*
    batas nilai toleransi (epsilon)
*/
const double TOLERANSI = 5e-15;
if(x < 0)
    return 0.0;

```

5. Menentukan nilai default y.

$x > 2 \rightarrow y = x/2$
 $x \leq 2 \rightarrow y = 1$

```

/*jika
    x > 2 -> y = x/2
    x <= 2 -> y = 1
*/
double y = (x > 2 ? x/2 : 1);

```

6. Ulangi sampai syarat dipenuhi : $\epsilon > |y*y - x|$, jika sudah memenuhi syarat, maka nilai akar adalah nilai y.

```

do{
    y = (y + x/y) / 2;
    cout << setw(10) << y << setw(12) << y*y << "\n";
}while(x > y*y + TOLERANSI || y*y > x + TOLERANSI);
return y;

```

Dari analisis di atas didapatkan source program sebagai berikut :

```
#include <iostream.h>
#include <iomanip.h>

double    babylon(double);

int main()
{
    double X;
    cout << "Input bilangan yang akan dicari akarnya : ";
    cin >> X;
    cout << "=====\n";
    cout << "\t Y \t Y*Y \n";
    cout << "=====\n";

    double y = babylon(X);
    cout << "=====\n";
    cout << "Akar " << X << " adalah : " << setprecision(20)
<<y <<

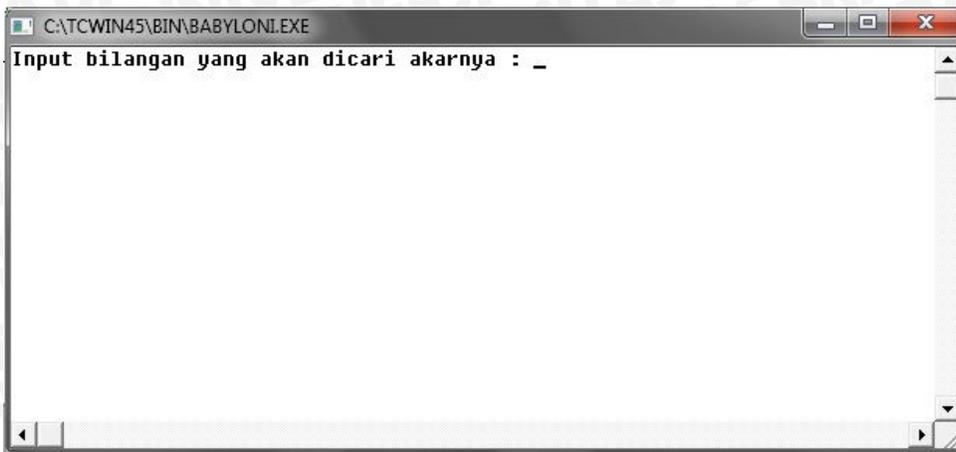
    endl;

    return 0;
}

double babylon(double x){

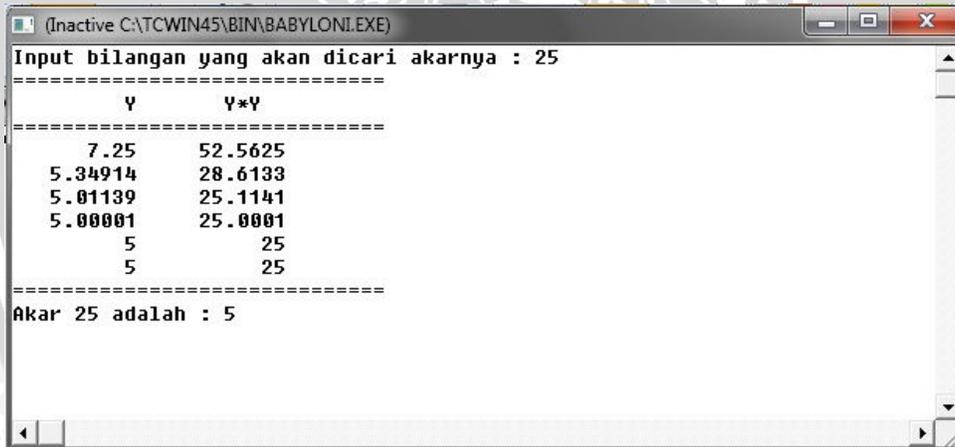
    const double TOLERANSI = 5e-15;
    if(x < 0)
        return 0.0;
```

Hasil program sebagai berikut :



Gambar 5.1 Antarmuka Program

Dengan masukan 25 sebagai contoh :

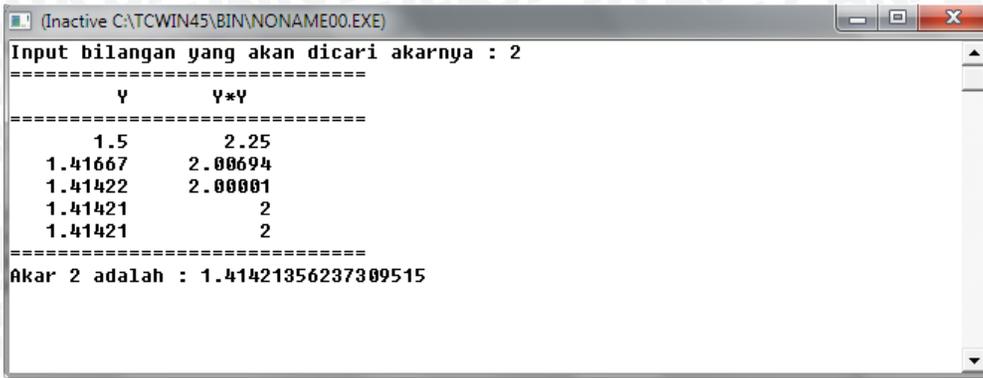


Gambar 5.2 Hasil Program

Analisi Pengujian

Pengujian aplikasi ini dilakukan dengan menggunakan beberapa sampel dengan masukan data yang berbeda :

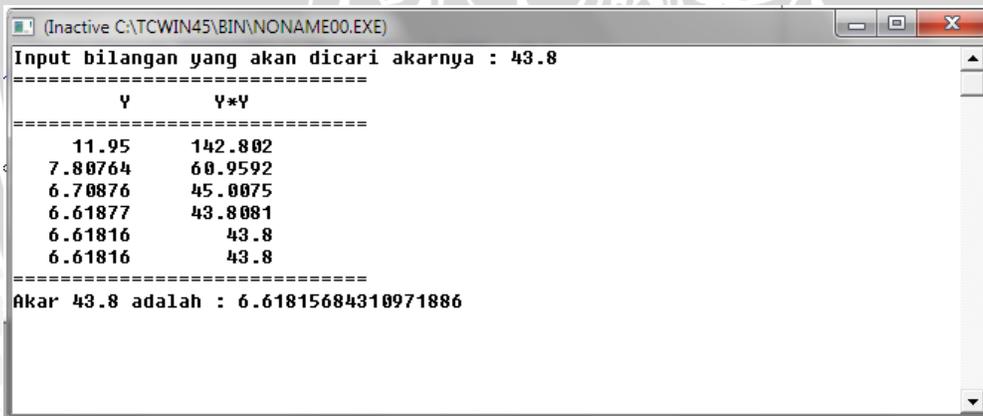
Dengan masukan data 2 :



Gambar 5.3 Program dengan Input Data 2

Pada masukan data 2 maka dibutuhkan 5 iterasi untuk mendapatkan hasil akarnya yaitu 1.41421356237309515

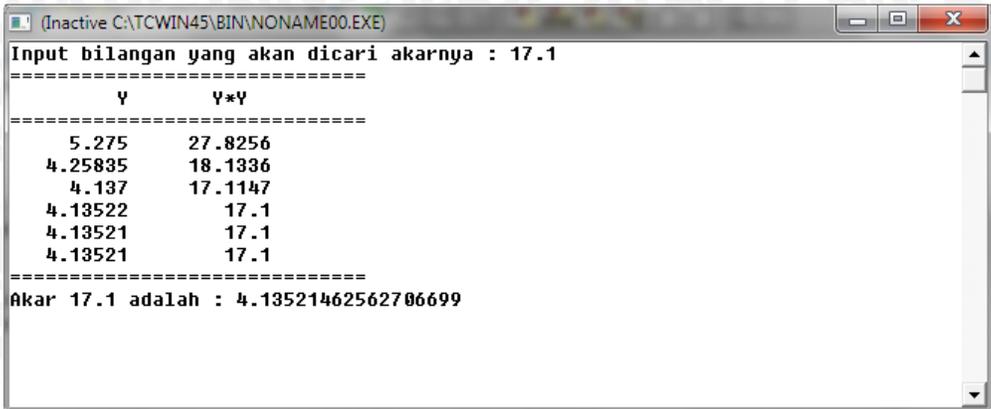
Dengan masukan data 43.8 :



Gambar 5.4 Program dengan Input Data 43.8

Pada masukan data 43.8 maka dibutuhkan 6 iterasi untuk mendapatkan hasil akarnya yaitu 6.61815684310971886

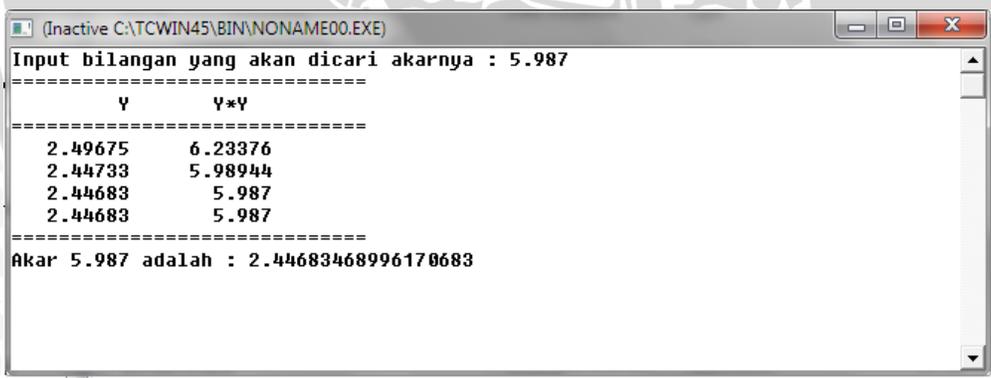
Dengan masukan data 17.1 :



Gambar 5.5 Program dengan Input Data 17.1

Pada masukan data 17.1 maka dibutuhkan 6 iterasi untuk mendapatkan hasil akarnya yaitu 4.13521462562706699

Dengan masukan data 5.987 :



Gambar 5.6 Program dengan Input Data 5.987

Pada masukan data 5.987 maka dibutuhkan 4 iterasi untuk mendapatkan hasil akarnya yaitu 2.44683468996170683

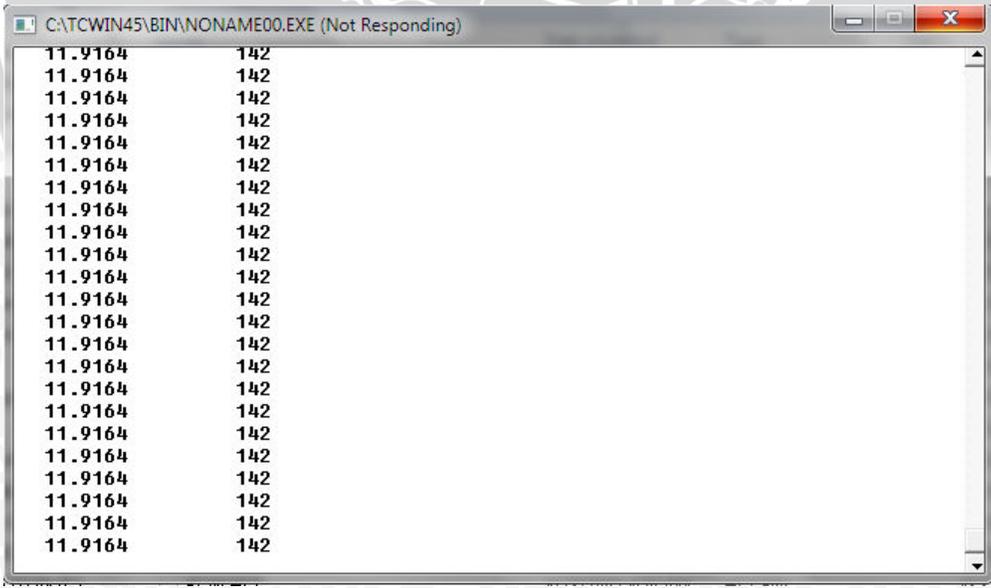
Dengan masukan data 197.3 :



Gambar 5.7 Program dengan Input Data 197.3

Pada masukan data 197.3 maka akar kuadratnya didapat 14.0464 dan program tidak berhenti juga.

Dengan masukan data 142 :



Gambar 5.8 Program dengan Input Data 142

Pada masukan data 142 maka akar kuadratnya didapat 11.9164 dan program tidak berhenti juga.

Berikut tabel hasil uji program :

Bilangan	Iterasi ke-n	Hasil Akar
1	7	1
3	6	1.732050808
5	6	2.236067977
7	7	2.645751311
9	6	3
11	7	3.31662479
13	8	3.605551275
15	8	3.872983346
17	8	4.123105626
19	8	4.358898944
21	8	4.582575695
23	8	4.795831523
25	7	5
27	8	5.196152423
29	8	5.385164807
31	8	5.567764363
33	8	5.744562647

35	9	5.916079783
37	9	6.08276253

Tabel 5.1 Hasil Uji Program (bulan)

Bilangan	Iterasi ke-n	Hasil Akar
1.3	8	1.140175425
2.5	7	1.58113883
5.8	7	2.408318916
7.4	7	2.720294102
9.6	7	3.098386677
11.5	8	3.391164992
13.9	8	3.728270376
15.7	8	3.962322551
17.6	8	4.195235393
19.1	8	4.370354677
213.7	10	14.61848145
235.6	10	15.34926708
257.9	10	16.05926524
273.2	10	16.52876281
291.9	10	17.08508121

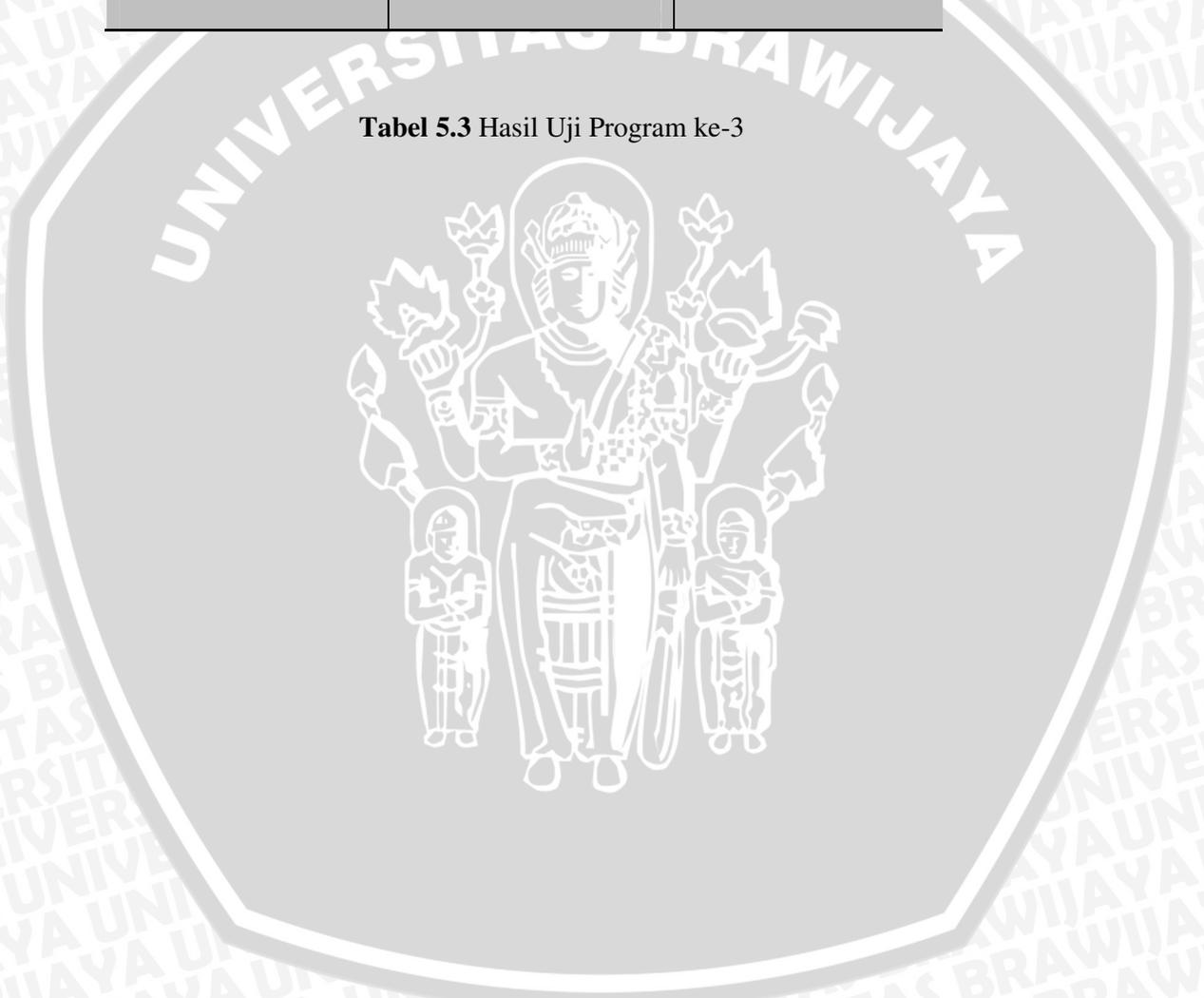
3134.1	12	55.98303314
3321.3	12	57.63072097
3573.5	12	59.77875877
3798.9	12	61.63521721

Tabel 5.2 Hasil Uji Program ke-2

Bilangan	Iterasi ke-n	Hasil Akar
1.37	8	1.170469991
3.51	6	1.8734994
5.85	7	2.418677324
7.43	7	2.725802634
9.68	7	3.111269837
11.523	8	3.394554463
13.934	8	3.732827347
15.765	8	3.970516339
17.678	8	4.204521376
19.198	8	4.381552236
21.7123	8	4.659645909
23.6342	8	4.861501826
25.9234	8	5.091502725

27.2234	9	5.217604814
29.9345	8	5.471243003
31.14566	9	5.580829687
33.32344	9	5.772645841
35.52346	9	5.960156038
37.94658	9	6.160079545

Tabel 5.3 Hasil Uji Program ke-3



BAB VI

Kesimpulan dan Saran

Berdasarkan hasil-hasil yang dicapai selama perancangan, pembuatan dan pengujian dari penelitian ini, maka dapat diambil beberapa kesimpulan dan saran.

6.1 Kesimpulan

1. Pada algoritma dengan metode babylonian masih memiliki banyak error sesuai dengan analisa dan hasil pengujian dimana metode ini hanya dapat digunakan dalam bilangan positif.
2. Dalam implementasinya algoritma pencarian akar kuadrat ini, pada bahasa pemrograman memiliki konvergensi error sangat besar.
3. Dengan metode atau algoritma yang digunakan maka untuk hasil error dapat dikurangi dengan setidaknya tetap memberikan 1 angka dibelakang koma untuk menjaga akurasi.

6.2 Saran

Dalam menggunakan metode pencarian akar kuadrat ini masih banyak algoritma yang dapat digunakan atau diterapkan pada program dan mempunyai error yang lebih sedikit

DAFTAR PUSTAKA

___http://en.wikipedia.org/wiki/Methods_of_computing_square_roots#Babylonian_method. 5 Juni 2012

___<http://mathlesstraveled.com/2009/05/18/square-roots-with-pencil-and-paper-the-babylonian-method/>. 12 Juni 2012

___<http://www.deltacollege.edu/dept/basicmath/Babylonian.htm>. 5 Juli 2012

Rachmat C., Antonius, 2010 Algoritma dan Pemrograman dengan Bahasa C, ANDI, Yogyakarta

Raharjo, Budi 2006 Pemrograman C++ rev.2, INFORMATIKA Bandung

Sutabri, Tata S.Kom., MM 2004 Pemrograman Terstruktur, Yogyakarta

Wikipedia. 2012. "**Algoritm**". <http://en.wikipedia.org/wiki/Algorithm>. 10 Juli 2012

Wikipedia. 2012. "**C (programming language)**". http://en.wikipedia.org/wiki/C_programing_language. 8 Juni 2012