

PEMBUATAN PERANGKAT LUNAK ALAT BANTU LOGIKA DAN ALGORITMA

TUGAS AKHIR

Diajukan untuk memenuhi persyaratan kelulusan studi
di

Joint Program D4 BA-Malang



Disusun oleh

GALUH PRASETYAWAN

NIM. 74 03 040 009

**JURUSAN TEKNOLOGI INFORMASI
JOINT PROGRAM D4 BA MALANG
MALANG 2007**

PEMBUATAN PERANGKAT LUNAK ALAT BANTU LOGIKA DAN ALGORITMA

Disusun Oleh :
Galuh Prasetyawan
NIM. 74 03 040 009



Telah disetujui dosen pembimbing
10 Mei 2006

Pembimbing :

Dosen Pembimbing I

Dosen Pembimbing II

Ali Ridho Barakbah, S.Kom.
NIP. 132 297 799

Abdul Munif S.ST
NIP. 132 302 287

PEMBUATAN PERANGKAT LUNAK ALAT BANTU

LOGIKA DAN ALGORITMA

Disusun Oleh :

Galuh Prasetyawan

NIM. 74 03 040 009

Telah Dipertahankan di Depan Panitia Penguji Ujian Tugas Akhir Mahasiswa Jurusan Teknik Elektro, Konsentrasi Teknologi Informasi, Program Diploma IV Joint Program BA-Malang, pada Tanggal 26 April 2007 dan Dinyatakan Telah Memenuhi Persyaratan Guna Memperoleh Gelar Sarjana Sains Terapan (S.ST)



Dosen Penguji I

Tita Karlita M.Kom
NIP. 132.300.680

Dosen Penguji II

Hero Yudo Martono ST.
NIP. 132 316 239

Dosen Penguji III

M. Rafie Pawellangi S.Pd.,S.St.
NIP. 132 302 032

ABSTRAKSI

Pembuatan perangkat lunak alat bantu logika dan algoritma , Galuh Prasetyawan, Ali Ridho Barakbah, S.Kom. dan Abdul Munif S.St .

Pada Jurusan Informatika semester awal, kita mendapatkan mata kuliah Logika dan Algoritma. Dalam mata kuliah ini kita akan mempelajari tentang algoritma beserta penerapannya dalam pemrograman. Dari permasalahan diatas penulis ingin memantu dalam mempermudah pembelajaran mata kuliah Logika dan Algoritma dengan membuat perangkat lunak alat bantu logika dan algoritma. Penulis akan menjelaskan bagaimana cara membuat aplikasi yang dapat menjadi alat bantu *Logika dan Algoritma*, menjelaskan faktor baik buruknya visualisasi, dan menjelaskan bagaimana membuat algoritma dengan baik dan benar.

Teori-teori yang dapat menunjang serta menjadi acuan dalam pembuatan proyek akhir ini dikelompokkan menjadi beberapa sub bab yaitu teori-teori *Visualisasi, Logika dan Algoritma*, masalah masalah beserta algoritmanya, *Struktur data, Flowchart*, dan sekilas tentang SwishMax.

Model pembuatan aplikasi yang digunakan adalah model *Waterfall*. Model waterfall berisi rangkaian aktivitas proses yang disajikan dalam proses yang terpisah, seperti spesifikasi kebutuhan, implementasi desain perangkat lunak, uji coba.

Kesimpulan yang dapat ditarik adalah langkah langkah dalam pembuatan perangkat lunak alat bantu logika dan algoritma adalah Analisa , Desain, Implementasi dan Pengujian. Yang mempengaruhi baik buruknya visualisasi adalah resolusi visualisasi, struktur data yang divisualisasikan, interaktifitas dari visualisasi, dan Computational infrastructure.

Kata kunci : visualisasi, *flowchart*, *algoritma*.

ABSTRACT

Making of software of appliance assist algorithm and logic , Galuh Prasetyawan, Ali Ridho Barakbah, S.Kom. dan Abdul Munif S.St .

At Majors of Informatics semester early, we get Logic and Algorithm. In eye of this was we will study concerning algorithm along with the applying of in pemrograman. Of above problems writer wish help in watering down study of eye of Logic and Algorithm made software of appliance assist algorithm and logic. Writer will explain how to make application able to become appliance assist Logic and Algorithm, explaining pros and cons factor of visualisation, and explain how to make algorithm better and correctness.

Theories able to support and also become reference in making of this final project grouped to become some chapter subs that is Visualisation theory, Logic and Algorithm, problem of problem along with his algorithm, Data structure, Flowchart, and in a flash concerning Swishmax.

Model making of application the used is model of Waterfall. Model waterfall contain presented process activity network in course of separate, like requirement specification, implementation of desain software, test-drive.

Conclusion of which can pulled by step do step in making of software of appliance assist algorithm and logic is Analysis , Desain, Implementation and Examination. influencing pros and cons of visualisation is visualisation resolution, data structure which isn't it, interaktifitas of visualisation, and Computational infrastructure.

Keywords : visualization, *flowchart*, *algorithm*.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT atas karunianya sehingga penulis dapat menyelesaikan Proyek Akhir dan menyusun buku ini dengan lancar. Rasanya tidak mungkin untuk menyelesaikan Proyek Akhir ini tanpa kerjasama dan dukungan dari banyak orang. Penulis juga mengucapkan terima kasih kepada Bapak Ali Ridho Barakbah dan Bapak Abdul Munif selaku Dosen Pembimbing atas bimbingannya selama ini.

Terima kasih penulis sampaikan pula untuk seluruh dosen di VEDC Malang, teman-teman seangkatan, dan semua pihak-pihak yang telah membantu yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari bahwa buku ini masih jauh dari sempurna. Oleh karena itu saran dan kritik yang membangun senantiasa penulis harapkan. Semoga apa yang penulis tuangkan dalam buku ini bermanfaat bagi semua orang yang membacanya.

Semoga karya ini bermanfaat bagi siapa saja yang tertarik untuk mempelajari visualisasi khususnya yang berhubungan dengan Logika dan Algoritma, dan tentunya agar dapat dikembangkan di masa mendatang.

Malang, 15 April 2007

Penyusun

UCAPAN TERIMA KASIH

Terselesaikannya Proyek Akhir ini tidak terlepas dari bantuan semua pihak di sekeliling penyusun. Oleh karena itu dengan segala kerendahan hati dan rasa hormat, penyusun mengucapkan terima kasih kepada :

1. Allah SWT atas berkat, rahmat serta hidayahNya, sehingga saya diberikan kemudahan untuk menyelesaikan proyek akhir ini.
2. Keluargaku tercinta yang senantiasa memberikan dorongan dan doa. Suatu anugerah yang luar biasa memiliki kalian semua dalam hidupku.
3. Bapak Abdul Munif dan Bapak Ali Ridho Barakbah , selaku dosen pembimbing. Terima kasih atas kesabaran dan perhatiannya selama pengerjaan Proyek Akhir ini.
4. Kepala Jurusan Teknologi Informasi , Wismanu Susetyo S.St. MT.
5. Pak Eko Subiyantoro S.Pd, S.St. yang telah memberikan judul ini.
6. Dosen – dosen gedung TI. *“Terima kasih untuk bimbingan dan arahan yang telah anda berikan selama ini.”*.
7. Teman – teman dari TI 3-1 dan TI 3-2 yang tidak bisa disebutkan satu persatu. *“Bisa mengenal kalian adalah suatu indahnya dunia”*.
8. Teman – teman jurusan lain dan semua pihak yang tidak bisa disebutkan satu persatu. *“Terima kasih banyak”*.
9. Seseorang yang selalu ada disampingku, *“Terima kasih banyak, Your Love Is Most Essential Thing In The Whole World”*.

DAFTAR ISI

ABSTRAKSI	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
UCAPAN TERIMA KASIH.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	ix
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	4
KAJIAN TEORI.....	5
2.1 Visualisasi	5
2.2 Flowchart.....	6
2.3 Logika dan Algoritma	7
2.3.1 Algoritma Bilangan Ganjil dan Genap.....	9
2.3.2 Algoritma Pengurutan 3 Bilangan.....	10
2.3.3 Algoritma Menghitung Keliling dan Luas Lingkaran.....	11
2.3.4 Algoritma Konversi Suhu Celcius ke Fahrenheit.....	12
2.3.5 Algoritma menampilkan bilangan ganjil antara 10-30.....	13
2.3.6 Algoritma menghitung determinan dan akar akar persamaan kuadrat.....	14
2.3.7 Algoritma tahun Kabisat	16
2.3.8 Algoritma menampilkan bilangan genap dari 2 sampai $< n$ kecuali bilangan genap tersebut kelipatan 4.....	17
2.3.9 Algoritma menghitung harga yang harus dibayar setelah mendapatkan diskon	18
2.3.10 Algoritma konversi jam ke menit.....	19
2.3.11 Algoritma konversi detik ke hari.....	20
2.3.12 Algoritma mencari max dan min dari deret bilangan.....	22
2.3.13 Algoritma kalkulator sederhana 2 bilangan yang terdiri dari operasi penjumlahan, pengurangan, perkalian, pembagian	22
2.3.14 Algoritma menampilkan tumpukan bilangan.....	23
2.3.15 Algoritma menghitung jumlah bola triangular.....	24
2.3.16 Algoritma menghitung angka angka dari suatu bilangan.....	25
2.3.17 Algoritma memilah suatu bilangan	26
2.3.18 Algoritma membalik kalimat	27
2.3.19 Algoritma menampilkan pembalikan suatu bilangan.....	29
2.3.20 Algoritma menentukan suatu kalimat termasuk kalimat palindrom atau bukan.....	30
2.3.21 Algoritma menghitung faktorial dari suatu bilangan	31
2.3.22 Algoritma menampilkan deret Fibonacci	32

2.3.23	Algoritma konversi desimal ke biner	33
2.3.24	Algoritma konversi biner ke desimal	34
2.3.25	Algoritma menentukan bilangan prima.....	35
2.4	Struktur Data	36
2.5	SwishMax.....	38
DESAIN DAN IMPLEMENTASI		43
3.1	Desain.....	43
3.1.1	Perancangan Arsitektur	43
3.1.2	Perancangan User Interface Menu	44
3.1.3	Perancangan User Interface Visualisasi	44
3.1.4	Perancangan Hardware dan Software untuk Server dan Client	45
3.2	Analisa Kebutuhan Visualisasi	46
3.3	Implementasi.....	48
3.3.1	Membuat Intro Visualisasi Algoritma Menghitung n Faktorial.....	50
3.3.2	Memberi penomoran pada flowchart	51
3.3.3	Membuat sprite untuk tiap chart	52
3.3.4	Membuat tampilan struktur data	53
3.3.5	Membuat tombol dan memberikan action.....	55
3.3.6	Memberi action pada setiap penomoran flowchart	55
PENGUJIAN PROGRAM DAN ANALISA.....		60
4.1	Pengujian.....	60
4.2.1	Pengujian Algoritma Bilangan Ganjil dan Genap.....	60
4.2.2	Pengujian Algoritma Mengurutkan 3 Bilangan	61
4.2.3	Pengujian Algoritma Menghitung Luas dan Keliling Lingkaran.....	62
4.2.4	Pengujian Algoritma Konversi Suhu Celcius Ke Fahrenheit.....	62
4.2.5	Pengujian Algoritma Bilangan Ganjil antara 10-30.....	63
4.2.6	Pengujian Algoritma Determinan dan Akar Persamaan kuadrat	63
4.2.7	Pengujian Algoritma Tahun Kabisat.....	64
4.2.8	Pengujian Algoritma n bilangan genap kecuali kelipatan 4.....	64
4.2.9	Pengujian Algoritma diskon.....	65
4.2.10	Pengujian Algoritma konversi jam ke menit.....	65
4.2.11	Pengujian Algoritma konversi detik ke hari.....	66
4.2.12	Pengujian Algoritma max min dari deret bilangan	66
4.2.13	Pengujian Algoritma kalkulator sederhana 2 bilangan	67
4.2.14	Pengujian Algoritma menampilkan tumpukan bilangan.....	67
4.2.15	Pengujian Algoritma jumlah bola triangular.....	68
4.2.16	Pengujian Algoritma menghitung angka angka dari bilangan	68
4.2.17	Pengujian Algoritma memilah suatu bilangan	69
4.2.18	Pengujian Algoritma membalik kalimat	69
4.2.19	Pengujian Algoritma membalik bilangan.....	70
4.2.20	Pengujian Algoritma kalimat palindrom.....	70
4.2.21	Pengujian Algoritma bilangan faktorial	71
4.2.22	Pengujian Algoritma deret fibbonaci	71
4.2.23	Pengujian Algoritma Desimal ke Biner	72
4.2.24	Pengujian Algoritma Biner ke Desimal	72
4.2.25	Pengujian Algoritma Bilangan Prima	73
4.2	Analisa Dan Hasil Uji Coba.....	73

KESIMPULAN DAN SARAN	74
5.1 Kesimpulan	74
5.2 Saran.....	74
DAFTAR PUSTAKA	75
BIODATA PENYUSUN.....	76

DAFTAR GAMBAR

Gambar 2.1 Flowchart algoritma bilangan Ganjil genap	7
Gambar 2.2 Flowchart algoritma pengurutan 3 bilangan	8
Gambar 2.3 Flowchart algoritma luas dan keliling lingkaran	9
Gambar 2.4 Flowchart algoritma konversi suhu celcius ke Fahrenheit	10
Gambar 2.5 Flowchart algoritma bilangan ganjil 10-30	11
Gambar 2.6 Flowchart algoritma akar persamaan kuadrat	12
Gambar 2.7 Flowchart algoritma tahun kabisat	13
Gambar 2.8 Flowchart algoritma bilangan genap kecuali kelipatan 4	14
Gambar 2.9 Flowchart algoritma diskon	15
Gambar 2.10 Flowchart algoritma konversi jam ke menit	16
Gambar 2.11 Flowchart algoritma konversi detik ke hari	17
Gambar 2.12 Flowchart algoritma max-min deret bilangan	18
Gambar 2.13 Flowchart algoritma kalkulator sederhana	19
Gambar 2.14 Flowchart algoritma tumpukan bilangan	20
Gambar 2.15 Flowchart algoritma bola triangular	21
Gambar 2.16 Flowchart algoritma menghitung angka dari suatu bilangan	21
Gambar 2.17 Flowchart algoritma memilah bilangan	22
Gambar 2.18 stack pada keadaan kosong	23
Gambar 2.19 mengisi stack dengan perintah Push	23
Gambar 2.20 Mengeluarkan isi stack dengan perintah Pop	23
Gambar 2.21 Flowchart algoritma membalik kalimat	24
Gambar 2.22 Flowchart algoritma membalik bilangan	25
Gambar 2.23 Flowchart algoritma kalimat palindrom	26
Gambar 2.24 Flowchart menghitung faktorial	27
Gambar 2.25 Flowchart deret Fibonacci	28
Gambar 2.26 Flowchart konversi bilangan decimal ke biner	29
Gambar 2.27 Flowchart algoritma konversi bilangan biner ke desimal	30
Gambar 2.28 Flowchart algoritma bilangan prima	31
Gambar 2.29 Tumpukan barang	32
Gambar 2.29 Menambah Script events	36
Gambar 3.1 Arsitektur Visualisasi Logika Algoritma	40
Gambar 3.2 Diagram Sistem Visualisasi Logika Algoritma	41
Gambar 3.3 Desain Menu Visualisasi Logika Algoritma	41
Gambar 3.4 Desain Gui Visualisasi Logika Algoritma	42
Gambar 3.5 Flowchart Algoritma Bilangan Faktorial	44
Gambar 3.6 Menu Visualisasi Logika Algoritma	45
Gambar 3.7 Gui Dari Algoritma Bilangan Faktorial	46
Gambar 3.8 Visualisasi Algoritma Bilangan Faktorial	46
Gambar 3.9 Diagram Alir Coding Visualisasi	47
Gambar 3.10 User Interface Dari Intro Visualisasi	48
Gambar 3.11 Jenis Effect Pada Intro Visualisasi	48
Gambar 3.12 Flowchart Yang Sudah Memiliki Penomoran	49
Gambar 3.13 Sprite Untuk Chart “Start”	49
Gambar 3.14 Efek Transform Pada Shape	50
Gambar 3.15 Struktur Data Bilangan Faktorial	51
Gambar 3.16 Objek Objek Dalam Scene 2	51

DAFTAR TABEL

Tabel 2.1 Flowchart algoritma bilangan Ganjil genap	33
Tabel 3.1 Jadwal pembuatan visualisasi algoritma bilangan faktorial	40
Tabel 4.1 Hasil Ujicoba	59

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Dalam matematika dan komputasi, algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma akan dapat selalu berakhir untuk semua kondisi awal yang memenuhi kriteria, berbeda dengan heuristik. Algoritma sering mempunyai langkah pengulangan (iterasi) atau memerlukan keputusan (logika Boolean dan perbandingan) sampai tugas selesai.

Desain dan analisis algoritma adalah suatu cabang khusus dalam ilmu komputer yang mempelajari karakteristik dan performa dari suatu algoritma dalam menyelesaikan masalah, terlepas dari implementasi algoritma tersebut. Dalam cabang disiplin ini algoritma dipelajari secara abstrak, terlepas dari sistem komputer atau bahasa pemrograman yang digunakan. Algoritma yang berbeda dapat diterapkan pada suatu masalah dengan kriteria yang sama. Kompleksitas dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Secara informal, algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki kompleksitas yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan kompleksitas yang tinggi.

Pada Jurusan Informatika semester awal, kita mendapatkan mata kuliah Logika dan Algoritma. Dalam mata kuliah ini kita akan mempelajari tentang algoritma beserta penerapannya dalam pemrograman. Kesulitan yang dihadapi dalam mata kuliah ini kita susah untuk mengerti algoritma atau penyelesaian dari suatu masalah yang diberikan oleh dosen serta sulitnya membayangkan struktur data yang akan digunakan. Dalam memahami penyelesaian suatu masalah, kita lebih mudah mengingat dan memahaminya apabila kita belajar melalui gambar atau visualisasi. Dari permasalahan diatas penulis ingin memantu dalam mempermudah pembelajaran mata kuliah Logika dan Algoritma dengan membuat perangkat lunak alat bantu logika dan algoritma.

1.2 RUMUSAN MASALAH

Agar penjelasan pada bab pembahasan akan lebih terarah maka penulis merumuskan masalah sebagai berikut :

1. Bagaimana langkah langkah membuat aplikasi yang dapat menjadi alat bantu *Logika dan Algoritma* ?
2. Apa saja yang mempengaruhi baik buruknya visualisasi?
3. Bagaimana menyusun algoritma dengan baik dan benar?

1.3 BATASAN MASALAH

Pada proyek akhir ini permasalahan difokuskan pada pembuatan visualisasi interaktif yang memvisualisasikan flowchart dari 25 algoritma yang sudah ditentukan yaitu:

1. Algoritma Bilangan Ganjil dan Genap
2. Algoritma Pengurutan 3 bilangan
3. Algoritma menghitung keliling dan luas lingkaran
4. Algoritma konversi suhu Celcius ke Fahrenheit
5. Algoritma menampilkan bilangan ganjil antara 10-30
6. Algoritma mencari determinan dan akar akar persamaan kuadrat
7. Algoritma Tahun Kabisat
8. Algoritma menampilkan bilangan genap dari 2 sampai $< n$ kecuali bilangan genap tersebut kelipatan 4.
9. Algoritma menghitung harga yang harus dibayar setelah mendapatkan diskon
10. Algoritma konversi dari jam ke menit
11. Algoritma konversi dari detik ke hari
12. Algoritma mencari max dan min dari suatu deret bilangan
13. Algoritma kalkulator sederhana 2 bilangan yang terdiri dari operasi penjumlahan, pengurangan, perkalian, pembagian.
14. Algoritma menampilkan tumpukan suatu bilangan
15. Algoritma menghitung jumlah bola triangular
16. Algoritma menghitung angka angka dari suatu bilangan
17. Algoritma memilah suatu bilangan
18. Algoritma membalik kalimat
19. Algoritma menampilkan pembalikan dari suatu bilangan

20. Algoritma mengecek suatu kalimat termasuk kalimat palindrom atau bukan
21. Algoritma menghitung faktorial dari suatu bilangan
22. Algoritma menampilkan deret Fibonacci
23. Algoritma konversi desimal ke biner
24. Algoritma binar biner ke desimal
25. Algoritma bilangan prima

Dalam membuat visualisasi ini penulis menggunakan aplikasi bantuan yaitu SwishMax.

1.4 TUJUAN

Kajian ini bertujuan menghasilkan suatu aplikasi yang dapat digunakan untuk memvisualisasikan *Logika dan Algoritma* yang diharapkan mempermudah dalam pembelajaran Mata kuliah *Logika dan Algoritma*.

1.5 METODOLOGI

1. Studi pustaka, yaitu pencarian tentang literatur tentang pengantar algoritma, sejarah algoritma, membuat flowchart dari beberapa kasus dan mempelajari tentang pembuatan animasi di SwishMax.
2. Perancangan sistem, yaitu merancang desain dari visualisasi modul ajar Logika dan Algoritma yaitu meliputi desain GUI (Graphical User Interface).
3. Melakukan rancang bangun visualisasi modul ajar Logika dan Algoritma.
4. Melakukan uji coba dengan metode Black-Box dan analisa terhadap visualisasi yang telah dibuat.

1.6 SISTEMATIKA PEMBAHASAN

Sistematika pembahasan proyek akhir yang disusun akan dibahas dalam bab-bab yang akan diuraikan di bawah ini :

BAB I Berisi tentang pendahuluan yang terdiri dari :

- Latar Belakang

- Rumusan Masalah
- Batasan Masalah
- Tujuan
- Metodologi
- Sistematika Pembahasan

BAB II Berisi tentang teori-teori penunjang yang terdiri dari :

- Visualisasi
- Logika dan Algoritma
- Permasalahan dan Algoritmanya
- Struktur Data
- Flowchart
- SwishMax

BAB III Perancangan Visualisasi Logika Algoritma dan Analisa kebutuhan sistem

BAB IV Berisi tentang pengujian dan analisa, membahas mengenai pengujian Visualisasi modul ajar Logika dan Algoritma.

BAB V Berisi tentang kesimpulan dan saran untuk proyek akhir, sehingga dapat dikembangkan menjadi lebih baik dimasa yang akan datang.

BAB II

KAJIAN TEORI

Dalam bab ini akan dibahas mengenai teori-teori yang dapat menunjang serta menjadi acuan dalam pembuatan proyek akhir ini. Bagian tersebut akan dikelompokkan menjadi empat sub bab yaitu teori-teori *Visualisasi*, *Logika dan Algoritma*, masalah masalah beserta algoritmanya, *Struktur data*, *Flowchart*, dan sekilas tentang SwishMax.

1.7 Visualisasi

Komponen visualisasi diperlukan karena dengan visualisasi kita bisa lebih mudah memahami, menganalisa persoalan yang terdiri dari data yang cukup banyak atau kompleks dan terperinci. Visualisasi merupakan suatu proses perubahan data, informasi, pengetahuan kepada suatu bentuk gambar yang bertujuan untuk mempermudah pemahaman terhadap data dan pengetahuan tersebut.

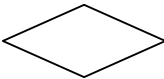
Ada beberapa faktor yang mempengaruhi baik tidaknya visualisasi^[9], yaitu

- ***Human perception***. Seseorang adalah konsumen dari visualisasi, sehingga faktor pembatas yang terakhir adalah "*human perception*". Pembuatan visualisasi suatu permasalahan dengan resolusi yang tinggi menjadi tidak bernilai dibanding dengan seseorang yang dapat melihat visualisasi dengan baik.
- ***Display resolution***. Visualisasi tampilkan dengan sebuah alat. Sehingga resolusi dari alat yang digunakan baik berupa monitor, *flat panel*, dinding, dapat diukur dengan *pixels*, *display size*, *display fidelity*.
- ***Interactivity***. Salah satu cara yang paling baik untuk meningkatkan kemampuan visualisasi adalah membuat visualisasi lebih interaktif. User ikut serta dalam jalannya visualisasi. Karena lebih interaktifnya user materi yang disampaikan lebih mudah dimengerti.
- ***Data structure and algorithm***. Struktur data dan algoritma yang efisien dibutuhkan untuk mencapai tampilan yang interaktif.
- ***Computational infrastructre***. Meliputi kecepatan CPU, *graphic card*, kapasitas penyimpanan.

1.8 Flowchart

Flowchart adalah gambaran dalam bentuk diagram alir dari algoritma dalam suatu program, yang menyatakan arah alur program dalam menyelesaikan suatu masalah.

Berikut adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart :

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Tabel 2.1 Bagian bagian dari flowchart

1.9 Logika dan Algoritma

Logika identik dengan masuk akal dan penalaran. Penalaran adalah salah satu bentuk pemikiran. Definisi logika sangat sederhana yaitu ilmu yang memberikan prinsip-prinsip yang harus diikuti agar dapat berfikir valid menurut aturan yang berlaku.

Pelajaran logika menimbulkan kesadaran untuk menggunakan prinsip-prinsip untuk berfikir secara sistematis.

Kata Algoritma berasal dari nama Abu Ja'far Mohammed Ibn Musa Alkhowarizmi, seorang ilmuwan Persia yang menulis buku berjudul Kitab Al jbr w'al-muqabala (rules of restoration and reduction) sekitar tahun 825. Sampai tahun 1950 istilah algorithm selalu diasosiasikan dengan Euclid's Algorithm, yaitu suatu proses yang menjelaskan cara mencari bilangan pembagi terbesar untuk dua buah bilangan^[2].

Pada *Merriam-Webster's Collegiate Dictionary*, istilah algoritma diartikan sebagai prosedur langkah demi langkah untuk memecahkan masalah atau menyelesaikan suatu tugas. Kamus Besar Bahasa Indonesia (KBBI) mendefinisikan algoritma sebagai urutan logis pengambilan keputusan untuk pemecahan masalah.

Menurut Donald E. Knuth^[7] sebuah algoritma harus memenuhi persyaratan:

1. *Finiteness*, Algoritma harus berakhir(terminate) setelah melakukan sejumlah langkah proses.
2. *Definiteness*, Setiap langkah algoritma harus didefinisikan dengan tepat dan tidak menimbulkan makna ambigu.
3. *Input*, Setiap Algoritma memerlukan data untuk diolah.
4. *Output*, setiap algoritma memberikan suatu output setelah mengerjakan proses.
5. *Effectiveness*, langkah langkah algoritma dikerjakan dalam waktu yang wajar

Langkah-langkah yang membentuk suatu algoritma dapat dibagi menjadi 3 kelompok proses yaitu :

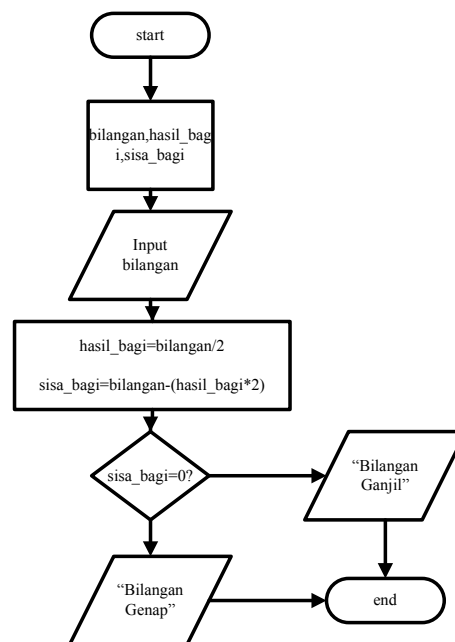
1. Sequence process, instruksi dikerjakan secara berurutan satu persatu mulai dari yang pertama sampai yang terakhir.
2. Selection process, suatu instruksi adakalanya baru boleh dikerjakan apabila memenuhi persyaratan tertentu misalnya bila pembayaran dilakukan secara tunai maka diberi diskon 10%, sedangkan pembelian dengan kredit tidak diberikan diskon. Dalam hal ini hanya boleh melakukan 1 instruksi dari dua alternatif instruksi (diskon=10% atau diskon =0).
3. Iteration process, suatu instruksi adakalanya perlu dikerjakan berulang ulang selama beberapa kali selama masih memenuhi suatu kondisi.

Dalam sub-bab dibawah ini akan dijelaskan beberapa masalah dengan algoritmanya yang dituangkan melalui flowchart.

Algoritma Bilangan Ganjil dan Genap

Bilangan-bilangan yang paling dikenal adalah bilangan bulat 0, 1, -1, 2, -2, ... dan bilangan-bilangan asli 1, 2, 3, ..., keduanya sering digunakan untuk berhitung dalam aritmatika. Himpunan semua bilangan bulat dalam buku-buku teks aljabar biasanya dinyatakan dengan lambang $\mathbb{Z}$ dan sedangkan himpunan semua bilangan asli biasanya dinyatakan dengan lambang $\mathbb{N}$.

Algoritma untuk menentukan suatu bilangan termasuk bilangan genap atau bilangan ganjil disajikan dengan flowchart dibawah ini :



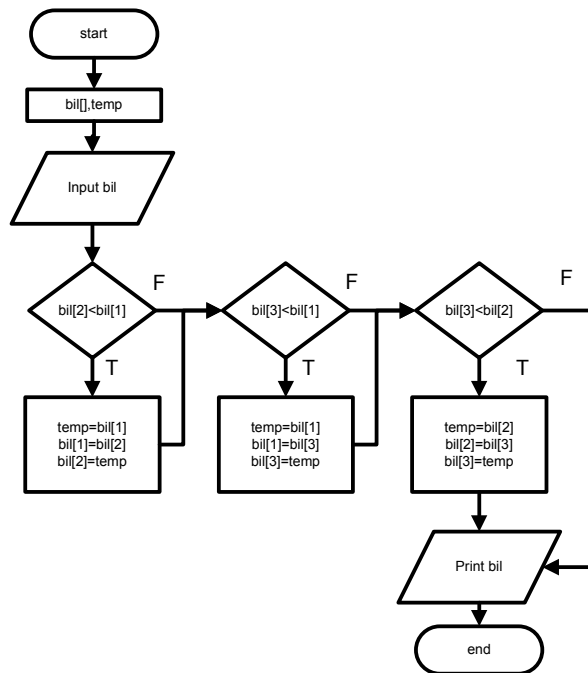
Gambar 2.1 Flowchart algoritma bilangan Ganjil genap

Bilangan genap adalah bilangan bulat yang habis (tidak mempunyai sisa) apabila dibagi 2. Sedangkan bilangan ganjil adalah bilangan bulat yang tidak habis apabila dibagi 2.

Algoritma Pengurutan 3 Bilangan

3 bilangan yang akan diurutkan dimasukkan dalam sebuah array. Metode pengurutan yang digunakan adalah membandingkan nilai dalam array dan penggunaan variable sementara.

Algoritma untuk mengurutkan 3 bilangan disajikan dengan flowchart dibawah ini :



Gambar 2.2 Flowchart algoritma pengurutan 3 bilangan

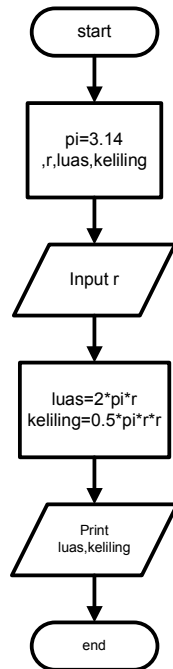
Terdapat 3 percabangan dalam algoritma pengurutan 3 bilangan diatas yaitu :

- Apabila $bil[2] < bil[1]$, maka $temp=bil[1]$, $bil[1]=bil[2]$, $bil[2]=temp$
- Apabila $bil[3] < bil[1]$, maka $temp=bil[1]$, $bil[1]=bil[3]$, $bil[3]=temp$
- Apabila $bil[3] < bil[2]$, maka $temp=bil[2]$, $bil[2]=bil[3]$, $bil[3]=temp$

Algoritma Menghitung Keliling dan Luas Lingkaran

Lingkaran adalah himpunan semua titik pada bidang dalam jarak tertentu yang disebut jari-jari dari suatu titik tertentu, yang disebut titik pusat. Lingkaran adalah contoh dari kurva tertutup sederhana, membagi bidang menjadi bagian dalam dan bagian luar.

Algoritma untuk menghitung luas dan keliling lingkaran disajikan dengan flowchart dibawah ini :



Gambar 2.3 Flowchart algoritma luas dan keliling lingkaran

Untuk menghitung luas dan keliling lingkaran menggunakan rumus :

$$\text{luas} = 2 * \pi * r$$

$$\text{keliling} = 0.5 * \pi * r * r$$

r = jari-jari

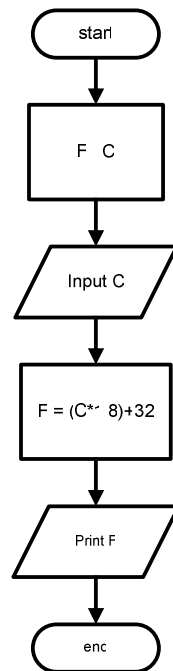
$\pi = 3.14$

Algoritma Konversi Suhu Celcius ke Fahrenheit

Derajat Celsius ($^{\circ}\text{C}$) adalah suatu satuan ukur suhu yang mendapatkan namanya dari ahli astronomi Anders Celsius (1701–1744), yang pertama kali mengusulkannya pada tahun 1742. Skala suhu Celsius didesain supaya titik beku air berada pada 0 derajat dan titik didih pada 100 derajat di tekanan atmosferik standar.

Fahreheit adalah salah satu skala temperatur selain Celsius dan kelvin. Nama Fahrenheit diambil dari ilmuwan Jerman yang bernama Gabriel Fahrenheit (1686-1736). Dalam skala ini, titik beku air adalah 32 derajat Fahrenheit (ditulis 32°F) dan titik didih air adalah 212 derajat Fahrenheit. Negatif 40 derajat Fahreheit sama dengan negatif 40 derajat Celsius.

Algoritma untuk konversi suhu Celcius ke Fahrenheit disajikan dengan flowchart dibawah ini :



Gambar 2.4 Flowchart algoritma konversi suhu celcius ke Fahrenheit

Untuk menghitung konversi suhu celcius ke fahrenheit menggunakan rumus :

$$F = (C * 1.8) + 32$$

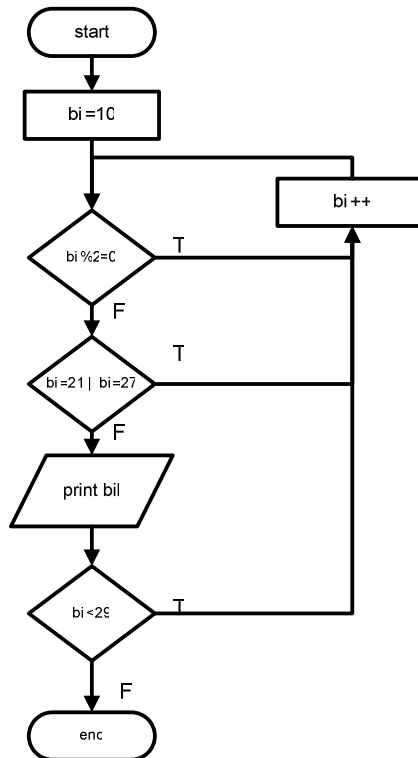
F= suhu dalam Fahrenheit

C=suhu dalam Celcius

Algoritma menampilkan bilangan ganjil antara 10-30

Bilangan ganjil antara 10 dan 30 adalah 11,13,15,17,19,21,23,25,27,29. Sedangkan yang akan ditampilkan adalah kecuali bilangan 21 dan 27. Jadi output yang diharapkan adalah 11,13,15,17,19,23,25,29.

Algoritma untuk menampilkan bilangan ganjil antara 10-30 kecuali bilangan 21 dan 27 disajikan dengan flowchart dibawah ini :



Gambar 2.5 Flowchart algoritma bilangan ganjil 10-30

Sebagai proses seleksi pertama agar program tidak mencetak bilangan 21 dan 27, diberikan syarat apakah bilangan = 21 atau bilangan = 27. Sebagai batas akhir supaya program tidak melebihi batas 30, maka diberikan syarat apakah bilangan < 29.

Algoritma menghitung determinan dan akar akar persamaan kuadrat

Persamaan kuadrat adalah suatu persamaan polinomial berorde dua. Bentuk umum dari persamaan kuadrat adalah :

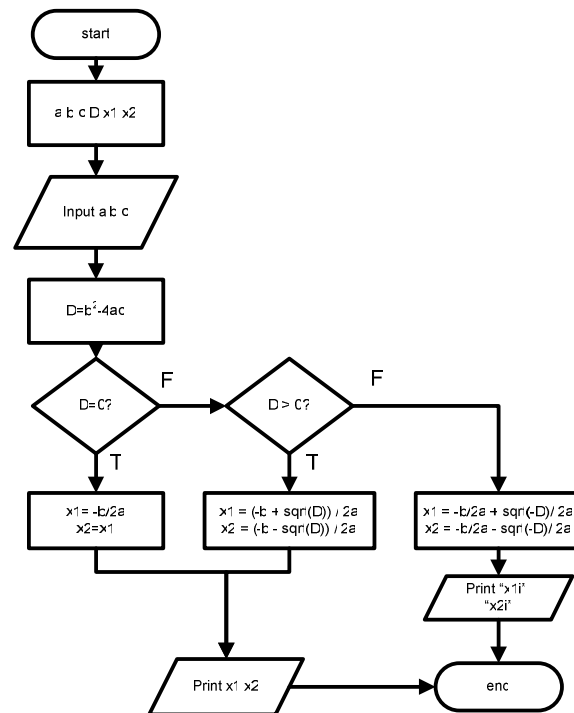
$$y = ax^2 + bx + c ; \text{dimana } a \neq 0$$

Huruf a , b dan c disebut sebagai koefisien: koefisien kuadrat a adalah koefisien dari x^2 , koefisien linier b adalah koefisien dari x , dan c adalah koefisien konstan atau disebut juga suku bebas.

Rumus kuadrat dikenal pula dengan nama '*rumus abc*' karena digunakan untuk menghitung akar-akar persamaan kuadrat yang tergantung dari nilai-nilai a , b dan c suatu persamaan kuadrat. Rumus yang dimaksud memiliki bentuk :

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Algoritma untuk menghitung determinan dan akar akar persamaan kuadrat disajikan dengan flowchart dibawah ini :



Gambar 2.6 Flowchart algoritma akar persamaan kuadrat

Determinan adalah nilai yang berada dalam akar (sqrt) pada rumus abc. Dalam hal ini diskriminan menentukan jumlah dan sifat dari akar-akar persamaan kuadrat. Terdapat tiga kasus yang mungkin:

1. Jika diskriminan bernilai nol, terdapat eksak satu akar, dan akar yang dimaksud merupakan bilangan riil. Hal ini kadang disebut sebagai akar ganda, di mana nilainya adalah:

$$x_1 = x_2 = -b/2a$$

2. Jika dikriminan bersifat positif, akan terdapat dua akar berbeda yang kedua-duanya merupakan bilangan riil.

$$x_1 = (-b - \text{sqrt}(D)) / 2a$$

$$x_2 = (-b + \text{sqrt}(D)) / 2a$$

3. Jika diskriminan bernilai negatif, *tidak* terdapat akar riil. Sebagai gantinya, terdapat dua buah akar kompleks (tidak-real).

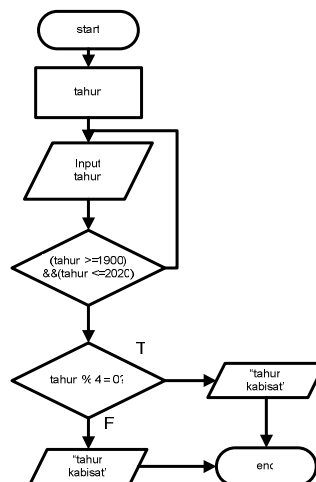
$$x_1 = -b / 2a + (\text{sqrt}(-D) / 2a) i$$

$$x_2 = -b / 2a - (\text{sqrt}(-D) / 2a) i$$

Algoritma tahun Kabisat

Tahun kabisat adalah tahun yang mempunyai 1 ekstra hari yang bertujuan supaya dalam kalender bisa sinkron dengan keadaan astronomi dan musim tahunan. Sebagai contoh, Bulan Februari mempunyai 29 hari pada tahun kabisat. Pada kalender Julian, setiap tahun yang bisa dibagi dengan 4 merupakan tahun kabisat. Tetapi pada kalender Gregorian, tahun yang bisa dibagi dengan 100 hanya dianggap sebagai tahun kabisat jika tahun ini juga bisa dibagi dengan 400. Misalkan tahun 1700, 1800 dan 1900 bukan tahun-tahun kabisat. Tetapi tahun 1600 dan 2000 merupakan tahun kabisat.

Algoritma untuk menentukan tahun kabisat disajikan dengan flowchart dibawah ini :



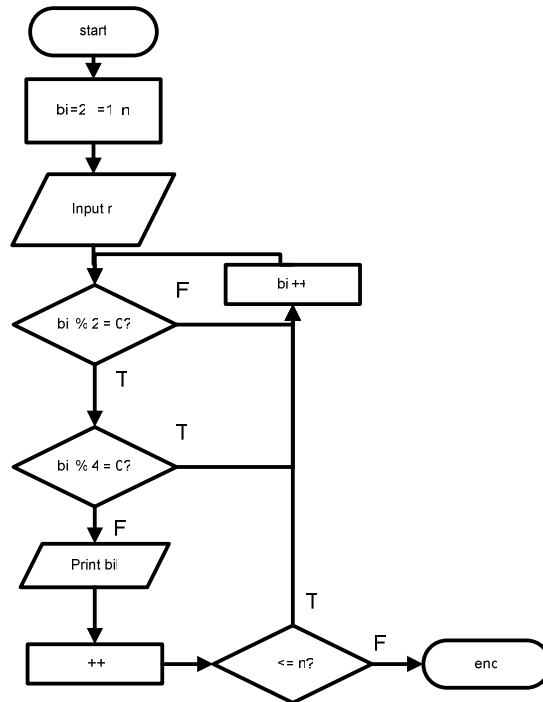
Gambar 2.7 Flowchart algoritma tahun kabisat

Algoritma diatas adalah untuk menentukan tahun kabisat pada kalender Julian saja, yaitu menggunakan syarat habis dibagi dengan 4 atau tidak. Selain itu batasan tahun yang boleh diinput adalah dalam rentang tahun 1900-2020.

Algoritma menampilkan bilangan genap dari 2 sampai < n kecuali bilangan genap tersebut kelipatan 4

Bilangan genap adalah bilangan bulat yang habis (tidak mempunyai sisa) apabila dibagi 2. Deret yang ditampilkan adalah deret bilangan genap dari 2 sampai ke n kecuali bilangan tersebut adalah kelipatan 4.

Algoritma untuk menampilkan deret bilangan genap dari 2 sampai ke n kecuali bilangan tersebut kelipatan 4 disajikan dengan flowchart dibawah ini :



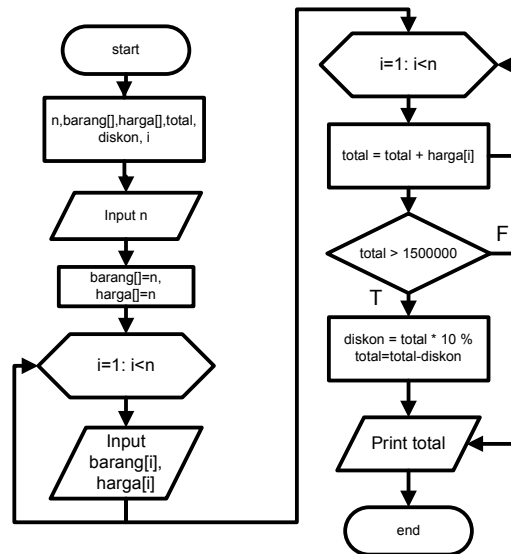
Gambar 2.8 Flowchart algoritma bilangan genap kecuali kelipatan 4

Misalkan input $n=5$, maka output deret yang ditampilkan adalah 2,6,10,14,18. Untuk menyeleksi bilangan genap menggunakan syarat $bil \% 2$ apakah mempunyai sisa apa tidak. Selain itu untuk menyeleksi bilangan kelipatan 4 menggunakan syarat $bil \% 4$ apakah mempunyai sisa apa tidak.

Algoritma menghitung harga yang harus dibayar setelah mendapatkan diskon

Algoritma untuk menghitung jumlah yang harus dibayar oleh pembeli setelah mendapatkan diskon 10% dengan syarat jumlah total pembelian < Rp.1.500.000,-

Algoritma untuk menghitung harga yang harus dibayar setelah mendapatkan diskon disajikan dengan flowchart dibawah ini :



Gambar 2.9 Flowchart algoritma diskon

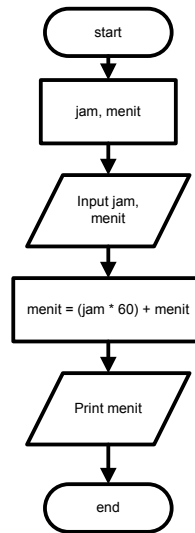
Jumlah barang bersifat dinamik sesuai input n dari user. Apabila jumlah total kurang dari 1500000 maka tidak akan mendapatkan diskon.

Algoritma konversi jam ke menit

Jam adalah sebuah unit waktu. Lama sebuah jam adalah 1/24 (satu perduapuluh empat) hari. Satu jam bisa dibagi menjadi unit waktu yang lebih kecil lagi yaitu menit dan detik.

Menit adalah sebuah unit waktu. Satu menit adalah 1/60 (seperenampuluh) jam. Satu menit terdiri dari 60 detik.

Algoritma untuk konversi dari jam ke menit disajikan dengan flowchart dibawah ini :



Gambar 2.10 Flowchart algoritma konversi jam ke menit

Rumus yang digunakan untuk mengkonversi dari jam ke menit adalah

$$\text{menit} = (\text{jam} * 60) + \text{menit}$$

Input dari user adalah format jam : menit. Jadi ada dua buah input dari user yaitu jam dan menit. Contoh apabila ingin mengkonversi 3 jam 45 menit , maka yang diinput pada jam adalah 3 dan menit adalah 45. Jadi

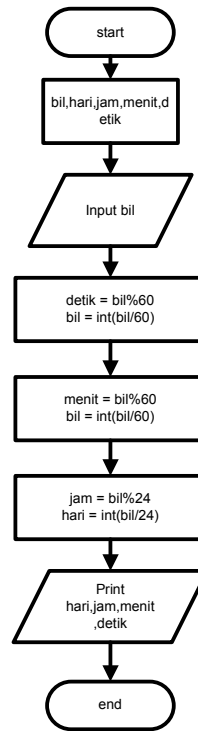
$$\text{menit} = (3 * 60) + 45$$

$$\text{menit} = 225$$

Algoritma konversi detik ke hari

Konversi detik ke hari bertujuan untuk merubah detik menjadi Hari, Jam, dan Menit. Menit adalah sebuah unit waktu. Satu menit adalah 1/60 (seperenampuluh) jam. Satu menit terdiri dari 60 detik. Jam adalah sebuah unit waktu. Satu jam adalah 1/24 hari. Satu jam terdiri dari 60 menit.

Algoritma untuk konversi dari detik ke hari disajikan dengan flowchart dibawah ini :



Gambar 2.11 Flowchart algoritma konversi detik ke hari

Contoh input bil = 11000, maka

Detik = bil % 60

$$= 11000 \% 60 = \mathbf{20}$$

bil = int(bil / 60)

$$= \text{int}(11000 / 60)$$

$$= 183$$

menit = bil % 60

$$= 183 \% 60 = \mathbf{3}$$

bil = int(bil / 60)

$$= \text{int}(183 / 60)$$

$$= 3$$

jam = bil % 24

$$= 3 \% 24 = \mathbf{0}$$

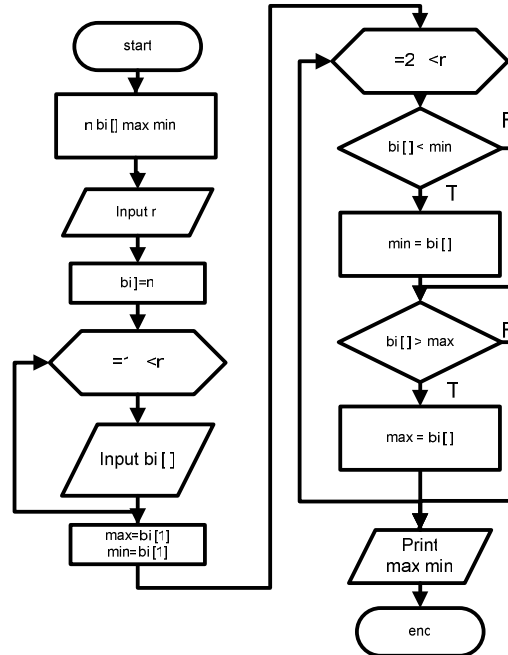
hari = int(bil / 24)

$$= \text{int}(3 / 24)$$

$$= 0$$

Algoritma mencari max dan min dari deret bilangan

Mencari nilai maks dan min dari suatu n deret bilangan yang diinput oleh user. Algoritma untuk mencari max-min dari suatu deret bilangan disajikan dengan flowchart dibawah ini :



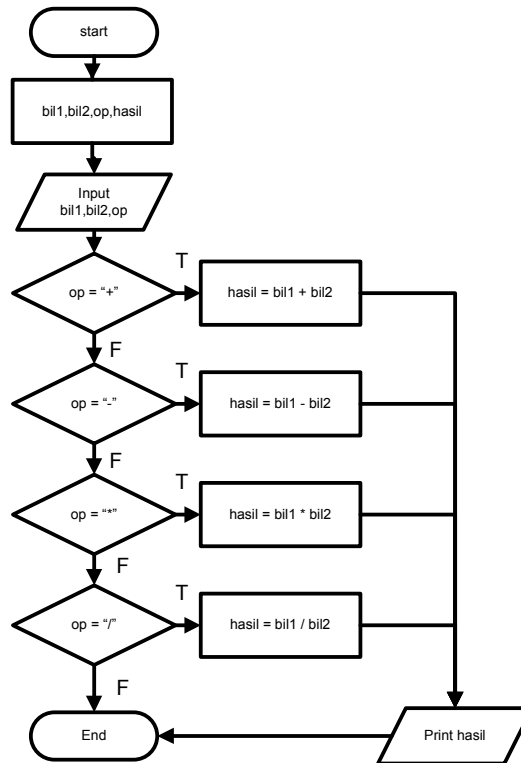
Gambar 2.12 Flowchart algoritma max-min deret bilangan

Jumlah deret dan nilai deret diinput oleh user. Sedangkan sebagai inisialisasi, max dan min diisi dengan nilai deret yang pertama. Contoh $n = 4$, $bi = 4, 8, 6, 7$, maka $max = 8$ dan $min = 4$.

Algoritma kalkulator sederhana 2 bilangan yang terdiri dari operasi penjumlahan, pengurangan, perkalian, pembagian

Kalkulator Sederhana yang dapat melakukan operasi +(penambahan), -(pengurangan), *(perkalian) dan /(pembagian). Kalkulator bersifat sederhana karena hanya melakukan perhitungan terhadap 2 bilangan saja.

Algoritma kalkulator sederhana disajikan dengan flowchart dibawah ini :



Gambar 2.13 Flowchart algoritma kalkulator sederhana

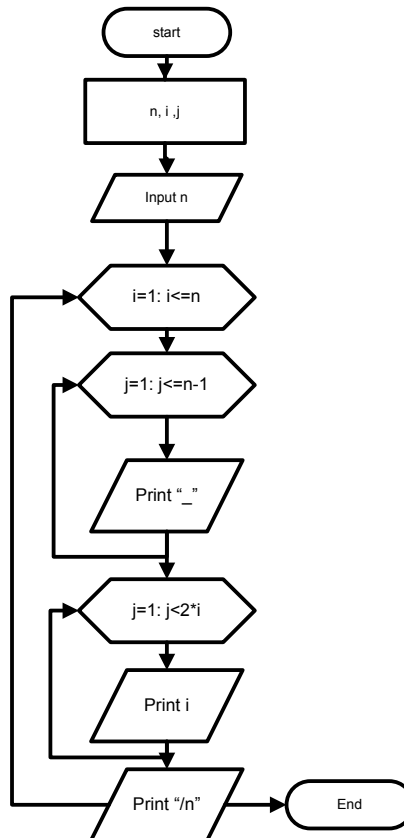
Contoh apabila bil1=7, bil2=4 dan op = “+” maka:

hasil = bil1+bil2

= 7+4=11

Algoritma menampilkan tumpukan bilangan

Algoritma untuk menampilkan segitiga tumpukan bilangan dengan batas sesuai input dari user. Algoritma tersebut disajikan dengan flowchart dibawah ini :



Gambar 2.14 Flowchart algoritma tumpukan bilangan

Contoh :

$n = 5$

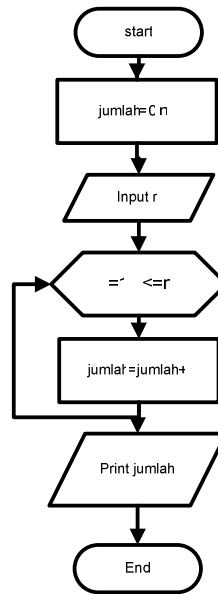
maka output tumpukan bilangan adalah seperti dibawah ini :

```

  1
 222
333333
44444444
5555555555
  
```

Algoritma menghitung jumlah bola triangular

Triangular adalah suatu susunan suatu benda(bola) yang disusun sedemikian sehingga menyerupai segitiga. Dengan mengetahui jumlah bola bagian bawah, maka dapat dihitung jumlah bola yang menyusun triangular tersebut. Algoritma menghitung jumlah bola triangular disajikan dengan flowchart dibawah ini :

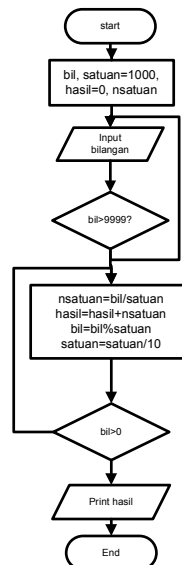


Gambar 2.15 Flowchart algoritma bola triangular

Contoh apabila $n = 3$ maka jumlah bola triangular adalah $3+2+1 = 6$ buah bola.

Algoritma menghitung angka angka dari suatu bilangan

Algoritma untuk menghitung angka-angka dari suatu bilangan yang diinput oleh user. Algoritma tersebut disajikan dengan flowchart dibawah ini :



Gambar 2.16 Flowchart algoritma menghitung angka dari suatu bilangan

Contoh :

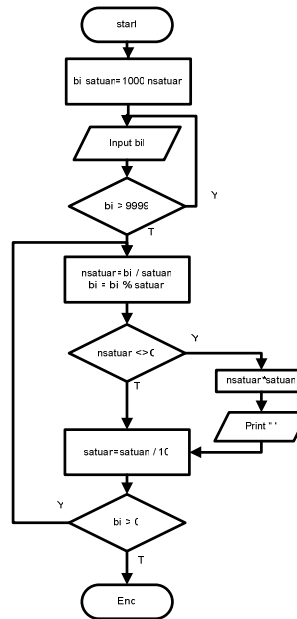
bilangan=3987

jumlah = $3+8+9+7 = 27$

Algoritma memilah suatu bilangan

Algoritma untuk memilah suatu bilangan maksimal bilangan 9999.

Algoritma tersebut disajikan dengan flowchart dibawah ini :



Gambar 2.17 Flowchart algoritma memilah bilangan

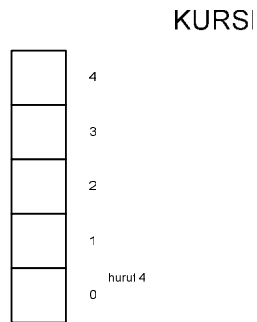
Contoh input bilangan = 1457

Output y = 1000, y = 400, y = 50, y=7

Algoritma membalik kalimat

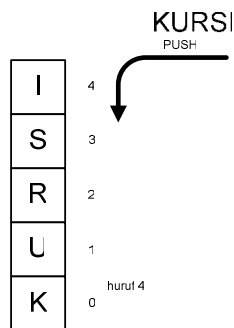
Algoritma untuk menampilkan kalimat dengan urutan terbalik. Struktur data yang digunakan adalah Stack. Untuk membalik bilangan, huruf huruf dari kalimat tersebut kita masukkan dalam stack menggunakan metode Push, setelah stack terisi, maka kita keluarkan kembali menggunakan metode Pop.

Pada algoritma mambalik kalimat, penggunaan struktur data stack diimplementasikan pada array. Dalam implementasinya pada array, kita harus menyiapkan sebuah array dengan panjang sama dengan jumlah huruf dalam kalimat yang akan dibalik.



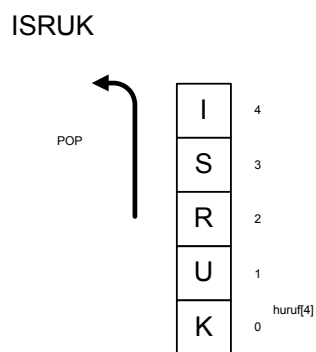
Gambar 2.18 stack pada keadaan kosong

Dalam gambar dibawah ini tiap huruf dari kalimat dimasukkan kedalam array dengan index ke-0 sampai ke-n dengan metode push.



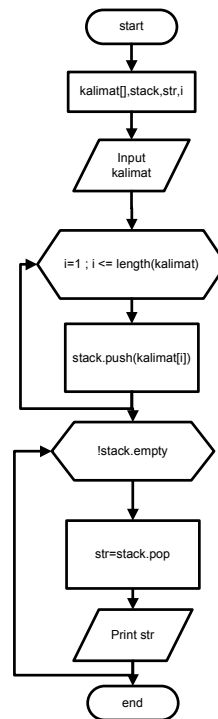
Gambar 2.19 mengisi stack dengan perintah Push

Lalu huruf huruf tersebut dikeluarkan mulai dari index ke-n sampai index ke-0.



Gambar 2.20 Mengeluarkan isi stack dengan perintah Pop

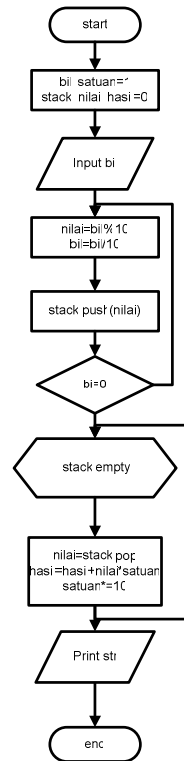
Maka kalimat yang dibalik menjadi kalimat dengan susunan huruf terbalik.



Gambar 2.21 Flowchart algoritma membalik kalimat

Algoritma menampilkan pembalikan suatu bilangan

Algoritma untuk menampilkan bilangan yang diinput oleh user dengan urutan terbalik. Algoritma yang digunakan bisa diimplementasikan sama dengan membalik kalimat. Tetapi dalam membalik bilangan ada cara yang lebih mudah yaitu dengan cara membagi dengan 10 dan menulis sisanya. Untuk lebih jelasnya dibawah ini adalah flowchart untuk membalik bilangan :



Gambar 2.22 Flowchart algoritma membalik bilangan

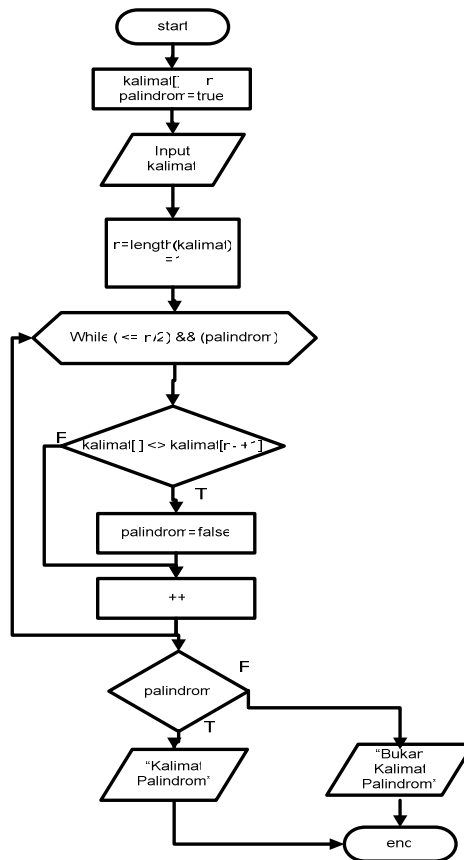
Contoh : input bil = 24

balik = $24 \% 10 = 4$ print balik

bil = $\text{int}(24/10) = 2$ print bil (karena $\text{bil} < 10$)

Algoritma menentukan suatu kalimat termasuk kalimat palindrom atau bukan

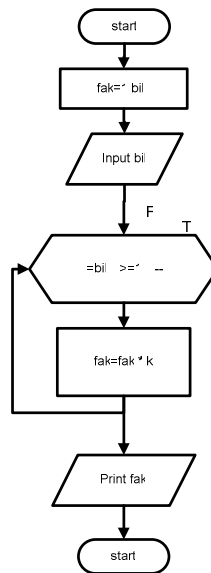
Kalimat palindrom adalah kalimat yang mempunyai urutan huruf yang sama dengan apabila kata tersebut dibalik. Contoh : “EKO OKE”. Dibawah ini adalah flowchart untuk menentukan kalimat tersebut adalah kalimat palindrom atau bukan.



Gambar 2.23 Flowchart algoritma kalimat palindrom

Algoritma menghitung faktorial dari suatu bilangan

Dalam matematika, faktorial dari bilangan asli n adalah hasil perkalian antara bilangan bulat positif yang kurang dari atau sama dengan n . Faktorial ditulis sebagai $n!$ dan disebut n faktorial. Algoritma tersebut disajikan dengan flowchart dibawah ini :



Gambar 2.24 Flowchart menghitung faktorial

Contoh untuk $n=6$, maka $6!=6*5*4*3*2*1 = 720$.

Algoritma menampilkan deret Fibonacci

Berdasarkan buku *The Art of Computer Programming* karya Donald E. Knuth, barisan ini pertama kali dijelaskan oleh matematikawan India, Gopala dan Hemachandra pada tahun 1150, ketika menyelidiki berbagai kemungkinan untuk memasukkan barang-barang ke dalam kantong. Di dunia barat, barisan ini pertama kali dipelajari oleh Leonardo da Pisa, yang juga dikenal sebagai Fibonacci (sekitar 1200), ketika membahas pertumbuhan ideal dari populasi kelinci.

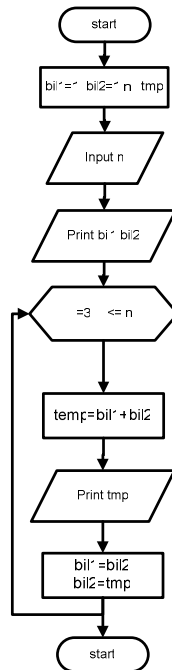
Dalam matematika, bilangan Fibonacci adalah barisan yang didefinisikan secara rekursif sebagai berikut:

$$F(0) = 1;$$

$$F(1) = 1;$$

$$F(n) = F(n-1) + F(n-2)$$

Algoritma untuk menampilkan deret Fibonacci disajikan dengan flowchart dibawah ini :



Gambar 2.25 Flowchart deret Fibonacci

Deretan ini berawal dari 1 dan 1, kemudian angka berikutnya didapat dengan cara menambahkan kedua bilangan yang berurutan sebelumnya. Dengan aturan ini, maka deretan bilangan Fibonacci yang pertama adalah:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946...

Algoritma konversi desimal ke biner

Algoritma untuk mengkonversi bilangan desimal yang diinput oleh user menjadi bilangan biner. Didalam dunia komputer kita mengenal tiga jenis bilangan, yaitu bilang biner,desimal dan hexadesimal. Bilangan biner atau binary digit adalah bilangan yang terdiri dari 1 dan 0. Sementara bilangan desimal adalah bilangan dari 0 sampai 9, dan yang terakhir adalah bilangan hexadesimal adalah bilangan 16 digit, dari 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Untuk mengkonversi bilangan decimal ke biner menggunakan cara :

Contoh bil = 45

$$45 / 2 = 22 \text{ sisa } 1$$

$$22 / 2 = 11 \text{ sisa } 0$$

$$11 / 2 = 5 \text{ sisa } 1$$

Membaca dari bawah
keatas

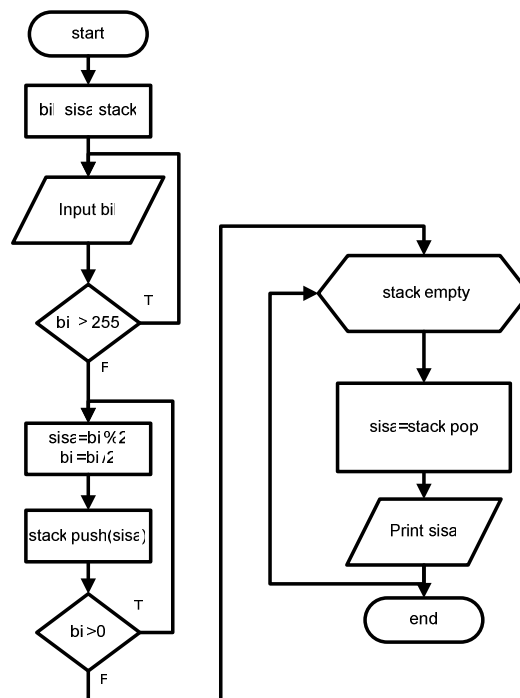
$$5 / 2 = 2 \text{ sisa } 1$$

$$2 / 2 = 1 \text{ sisa } 0$$

$$1 / 2 = 0 \text{ sisa } 1$$

Jadi bilangan biner dari 45 adalah 101101

Algoritma untuk mengkonversi bilangan decimal ke biner diatas diterjemahkan dengan flowchart dibawah ini :



Gambar 2.26 Flowchart konversi bilangan decimal ke biner

Algoritma konversi biner ke desimal

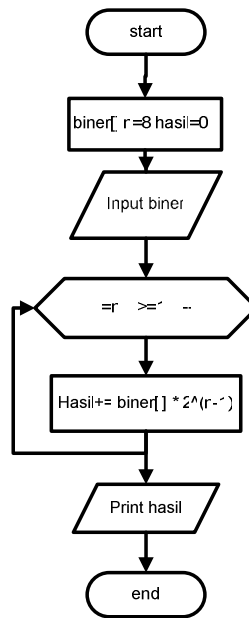
Algoritma untuk mengkonversi bilangan biner yang diinput oleh user menjadi bilangan desimal. Cara untuk mengkonversi bilangan biner ke decimal adalah sebagai berikut :

Contoh bil = 1010101

1	0	1	0	1	0	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	0	16	0	4	0	1

$1 \Rightarrow 64 + 0 + 16 + 0 + 4 + 0 + 1 = 85$

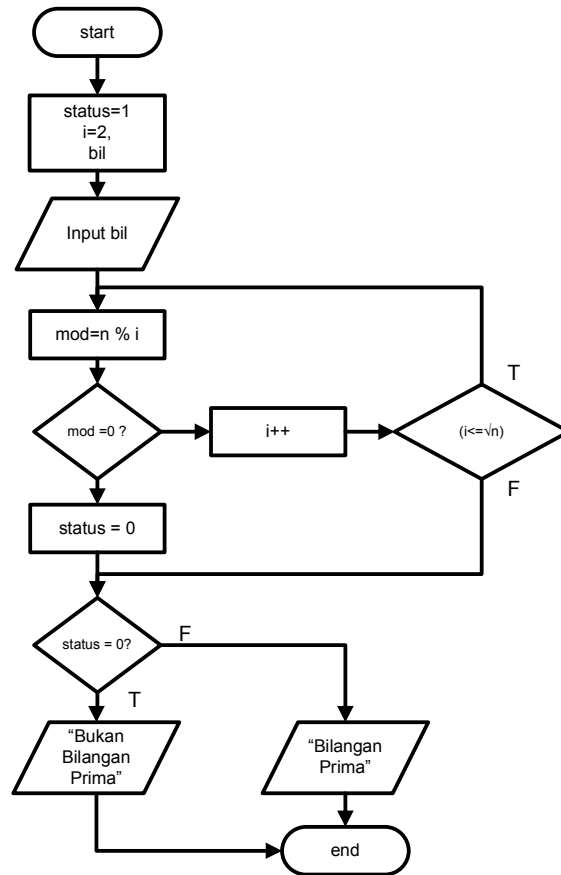
Algoritma untuk mengkonversi bilangan biner ke desimal diatas diterjemahkan dengan flowchart dibawah ini :



Gambar 2.27 Flowchart algoritma konversi bilangan biner ke desimal

Algoritma menentukan bilangan prima

Dalam matematika, bilangan prima adalah bilangan asli yang lebih besar dari 1, yang faktor pembaginya adalah 1 dan bilangan itu sendiri. Sepuluh bilangan prima yang pertama adalah 2, 3, 5, 7, 11, 13, 17, 19, 23 dan 29. Jika suatu bilangan yang lebih besar dari satu bukan bilangan prima, maka bilangan itu disebut bilangan komposit. Algoritma untuk menentukan bilangan prima atau bukan disajikan dalam flowchart^[4] dibawah ini :



Gambar 2.28 Flowchart algoritma bilangan prima

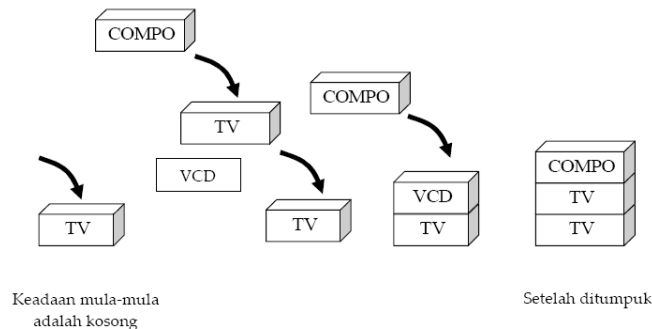
1.10 Struktur Data

Dalam istilah ilmu komputer, sebuah struktur data adalah cara penyimpanan, pengorganisasian dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien.

Beberapa contoh struktur data adalah :

- Variabel, sebagai tempat untuk menyimpan suatu nilai yang sejenis. Terdiri dari nama dari variable itu sendiri dan nilai yang disimpan.
- Array, sebuah struktur data yang terdiri atas banyak variabel dengan tipe data sama, dimana masing-masing elemen variabel mempunyai nilai indeks. Setiap elemen array mampu untuk menyimpan satu jenis data (yaitu: variabel) ^[5].
- Matriks, sebagai perwujudan dari array dua dimensi, operasi aritmatika seperti penjumlahan, perkalian, dan pengurangan bisa dilakukan.

- Stack, mekanisme penyimpanan seperti tumpukan benda. Konsep utamanya adalah LIFO (Last In First Out), benda yang terakhir masuk dalam stack akan menjadi benda pertama yang dikeluarkan dari stack^[5].



Gambar 2.29 Tumpukan barang

Pada gambar di atas, jika kita ingin mengambil sesuatu dari tumpukan maka kita harus mengambil benda paling atas dahulu, yakni compo. Misalnya jika VCD langsung diambil, compo akan jatuh. Prinsip stack ini bisa diterapkan dalam pemrograman. Di C++, ada dua cara penerapan prinsip stack, yakni dengan array dan linked list.

- Queue, Jika diartikan secara harafiah, queue berarti antrian, queue merupakan salah satu contoh aplikasi dari pembuatan double linked list yang cukup sering kita temui dalam kehidupan sehari-hari, misalnya saat anda mengantri di loket untuk membeli tiket. Istilah yang cukup sering dipakai seseorang masuk dalam sebuah antrian adalah enqueue. Dalam suatu antrian, yang datang terlebih dahulu akan dilayani lebih dahulu. Istilah yang sering dipakai bila seseorang keluar dari antrian adalah dequeue.

1.11 SwishMax

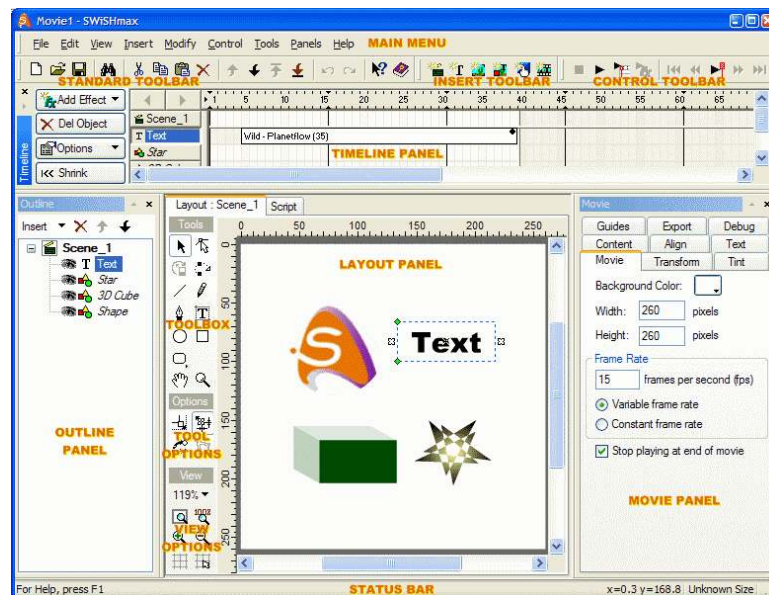
Saat ini telah bermunculan program-program animasi diantaranya adalah Macromedia Flash dan SwishMax. Animasi yang dihasilkan dari program-program ini adalah berformat Flash movie (swf). Animasi-animasinya ini lebih sering digunakan untuk movie/film, banner iklan, button, navigasi atau presentasi, baik dalam sebuah homepage maupun berdiri sendiri. Program ini dapat didownload secara gratis versi trialnya di <http://www.swishzone.com>.

Dari beberapa program animasi yang ada, SwishMax lah yang paling mudah untuk digunakan dan mungkin juga telah menarik minat banyak kaum flasher maupun web developer. Karena, Swishmax memiliki feature yang cukup untuk menghasilkan animasi komplek yang indah, baik animasi teks, image, grafik dan suara. Selain itu Script nya pun mudah dipelajari, karena adanya menu guided yang disediakan untuk pemula.

SwishMax memiliki tampilan antarmuka/ruang kerja yang cukup bagus, yang terdiri dari beberapakomponen, yaitu :

- Main menu
- Tollbox
- Tollbar-toolbar termasuk toolbar Standar, insert, dan control
- Panel-panel termasuk panel Timeline, Outline, Layout, dan Text
- Status Bar

Setiap komponen tersebut diatas masing-masing memiliki berbagai macam fungsi dan fasilitas yang mendukung dalam pembuatan animasi Flash. Gambar berikut menampilkan masing-masing komponen yang telah disebutkan diatas .



Gambar 2.30 Tampilan dari Swishmax

Sebelum kita belajar lebih jauh tentang SwishMax, perlu kita ketahui dulu elemen-elemen dasar dari sebuah animasi Flash yang dibentuk dari SwishMax. Hal pertama yang harus kita pahami adalah animasi yang dibuat dengan SwishMax akan kita

sebut dengan movie. Sedangkan movie itu sendiri terdiri dari satu atau lebih scene, dan setiap scene memiliki satu atau lebih timeline. Timeline adalah kumpulan beberapa frame. Dari setiap timeline dari sebuah scene dapat kita tepatkan objek, dan objek itu dapat pula kita beri efek. Efek-efek akan mulai dan berhenti pada frame tertentu dan dapat dikontrol melalui Script. Untuk membuat sebuah movie baru, kita tinggal menekan tombol File > New atau menekan bersamaan tombol CTRL dan tombol N.

Scene adalah kumpulan dari beberapa objek yang telah diberi efek-efek tertentu dalam sebuah timeline. Scene baru dapat dibuat dengan menekan tombol Insert > Scene. Untuk menambahkan sebuah objek ke dalam Scene, kita cukup menekan salah satu objek yang ada pada Tollbox, kemudian klik atau drag pada Layout Panel dimana kita akan menempatkan objek tersebut. Seperti halnya sebuah film biasa, Scene terdiri atas beberapa buah frame yang akan berjalan berurutan ataupun tidak yang dapat dikontrol melalui perintah pada panel Script. Sedangkan Timeline merupakan kumpulan dari frame-frame, yang dimulai dari frame sebelah kiri dan berakhir pada frame sebelah kanan.

SwishScript merupakan pengganti dari Panel Action pada Swish v.2. Swishscript adalah bahasa yang digunakan dalam SwisHmax untuk memprogram sebuah movie dan item-itemnya dalam sebuah movie. SwisHscript dapat digunakan untuk :

- mendefinisikan Action-action yang akan terjadi pada suatu frame tertentu, ketika dua buah
- obyek bertemu/bertabrakan, atau ketika kita akan meng-input variable dari file external.
- Mengontrol dalam pemakain suara/sound
- Berinteraktif dengan file-file external seperti PHP, ASP ataupun Text
- Untuk menifinisikan property fisik dari sebuah obyek dan lain-lainnya

SwisHscript inilah bagian yang sangat penting dan menarik dari SwisHmax, karena dengan adanya script ini banyak yang dapat kita lakukan seolah-olah kita sedang menggunakan suatu bahasa pemrograman. Ada dua mode yang diberikan , yaitu mode “Guide” yang menampilkan action-action yang umum dan sering digunakan, dan mode “Expert” yang semua action diketik sendiri. Untuk pemula lebih baik menggunakan mode guide.

Seperti halnya dalam suatu bahasa pemrograman, SwisHscript juga mengenal istilah yang kita sebut dengan “variable”. Variabel biasanya digunakan untuk

menyimpan suatu nilai, baik berupa masukan ataupun nilai keluaran/hasil, seperti contoh berikut :

```
x = 3 + 1
```

maka variabel “x” akan tersimpan dengan nilai (3+1) atau 4. Jika kita lihat contoh berikut :

```
y = x * x // y = 16
```

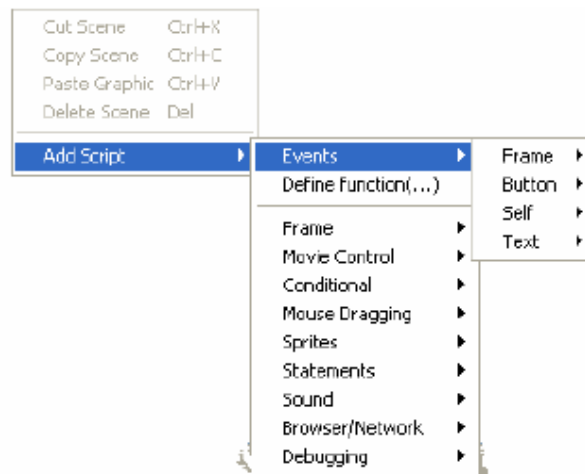
```
x = x + 1 // x = 5, dimana nilai x sebelumnya adalah 4
```

Perlu diperhatikan, dalam Swishscript tanda // digunakan sebagai area komentar.

Variabel juga dapat digunakan untuk menyimpan nilai berupa teks (variable string) seperti contoh berikut :

```
Pesan = "selamat pagi" // pesan= "selamat pagi"
```

Ada beberapa hal yang perlu dipelajari dari scripting ini yaitu tentang Event dan Actions. Semua Actions dibangkitkan dari sebuah Event. Event dapat terjadi ketika movie mencapai suatu frame tertentu, respon dari mouse ataupun keyboard. Sebuah Event dapat memicu lebih dari satu actions. Untuk memudahkan pemahamannya kita contohkan sebagai berikut : Ketika movie mencapai sebuah frame atau ketika mouse melewati suatu obyek (event), movie berhenti (Actions Stop) dan memainkan sebuah .mp3 file (actions play sound). Untuk menambahkan Frame events pada suatu movie, pilihlah frame dari sebuah obyek, kemudian pilih tab “Script” pada “Panel Layout”, klik kanan dan pilih Add Script > Events > Frame seperti gambar berikut :



Gambar 2.29 Menambah Script events

Ada empat tipe dari events, yaitu :

1. Frame Events : terjadi ketika movie mencapai frame tertentu, diantaranya adalah :

- a. **onFrame** : Event akan dijalankan saat pada frame tertentu.
 - b. **onLoad** : Event akan dijalankan ketika Sprite atau Scene dijalankan pertama kali.
 - c. **onEnterFrame** : event akan dijalankan pada setiap frame yang dilalui
2. **Self Events** : terjadi ketika mouse berinteraksi dengan sebuah objek atau saat menekan sebuah tombol pada keyboard. Hanya obyek yang didefinisikan sebagai target (“Target” dicentang) yang dapat menggunakan event ini. Macam-macam event ini adalah :
- a. **onSelfEvent (press)** : menekan klik kiri mouse saat kursor diatas obyek
 - b. **onSelfEvent (release)** : melepaskan tombol kiri mouse saat kursor diatas obyek
 - c. **onSelfEvent (roll over)** : memindahkan kursor mouse dari luar obyek ke atas obyek tanpa menekan mouse
 - d. **onSelfEvent (roll out)** : memindahkan kursor mouse dari obyek ke luar obyek tanpa menekan mouse
 - e. **onSelfEvent (drag over)** : memindahkan kursor mouse dari luar obyek ke atas obyek dengan kondisi tombol kiri mouse ditekan
 - f. **onSelfEvent (drag out)** : memindahkan kursor mouse dari obyek ke luar obyek dengan kondisi tombol kiri mouse ditekan
 - g. **onSelfEvent (release out)** : melepaskan tombol kiri mouse setelah memindahkan kursor keluar dari obyek
 - h. **onSelfEvent (keypress)** : saat tombol pada keyboard ditekan.
3. **Button Events** : terjadi ketika mouse berinteraksi dengan sebuah objek atau saat menekan sebuah tombol pada keyboard. Penambahan Button Event pada sebuah obyek akan membuat obyek tersebut diperlakukan seperti sebuah Button/tombol. Events ini mirip sekali dengan Selt Event, sehingga penulis tidak mendiskripsikannya satu-persatu lagi.
4. **Text Event** yang hanya terdiri dari satu event yaitu “onChange”, berlaku jika ada perubahan pada suatu text.

Actions adalah sebuah perintah yang dipicu oleh sebuah event. Action dapat mengubah permainan dari movie, memainkan atau menghentikan suara/sound, memanggil movie atau halaman web lain, atau berkomunikasi dengan browser host atau player. Property Action berada bersamaan dengan Event, dengan mengklik kanan tombol mouse pada sebuah event. Berikut adalah action-action yang ada pada mode Guide antara lain :

- **Stop** : digunakan untuk menghentikan movie atau sprite yang sedang dimainkan.
- **Goto and Play** : digunakan untuk menuju frame tertentu. Frame ini dapat berada dalam scene yang sama, scene lain atau dalam sprite.
- **nextSceneAndPlay** : digunakan untuk menuju scene tertentu selanjutnya akan memainkan frame pertama dari scene tersebut.

BAB III

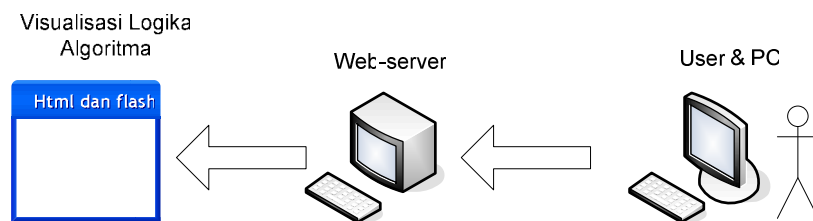
DESAIN DAN IMPLEMENTASI

1.12 DESAIN

Dalam perencanaan system terdiri dari beberapa perencanaan yaitu perencanaan Arsitektur dan perencanaan GUI (Graphical User Interface).

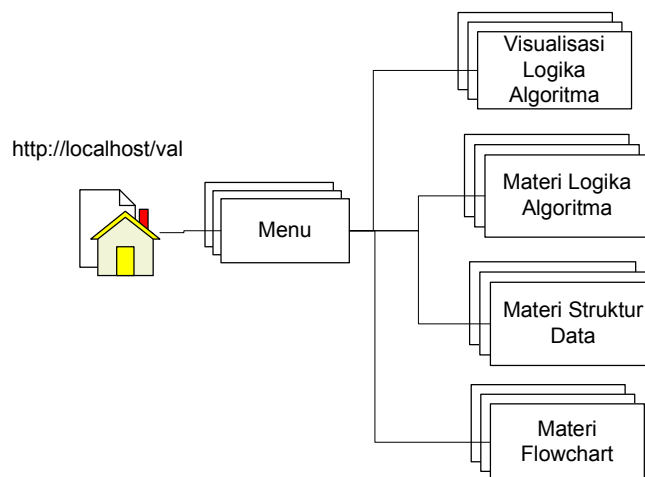
1.12.1 Perancangan Arsitektur

Arsitektur sistem visualisasi logika dan algoritma adalah sebagai berikut :



Gambar 3.1 Arsitektur Visualisasi Logika Algoritma

Visualisasi yang telah dibuat menggunakan SwishMax diexport dalam bentuk *.swf lalu di-embed dalam file html lalu disimpan dalam sebuah webserver. Tujuan dari penggunaan web sebagai media visualisasi adalah lebih mudahnya delivery content ke user pemakai. Sedangkan diagram sistem visualisasi logika algoritma seperti gambar dibawah :

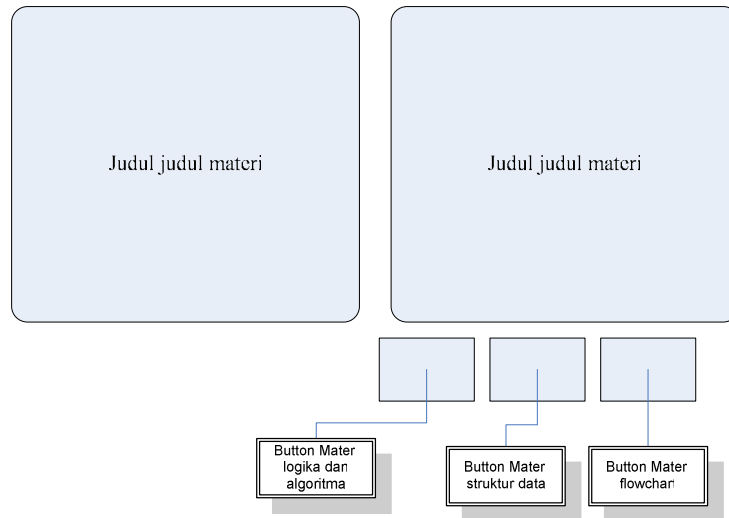


Gambar 3.2 Diagram sistem visualisasi logika algoritma

1.12.2 Perancangan User Interface Menu

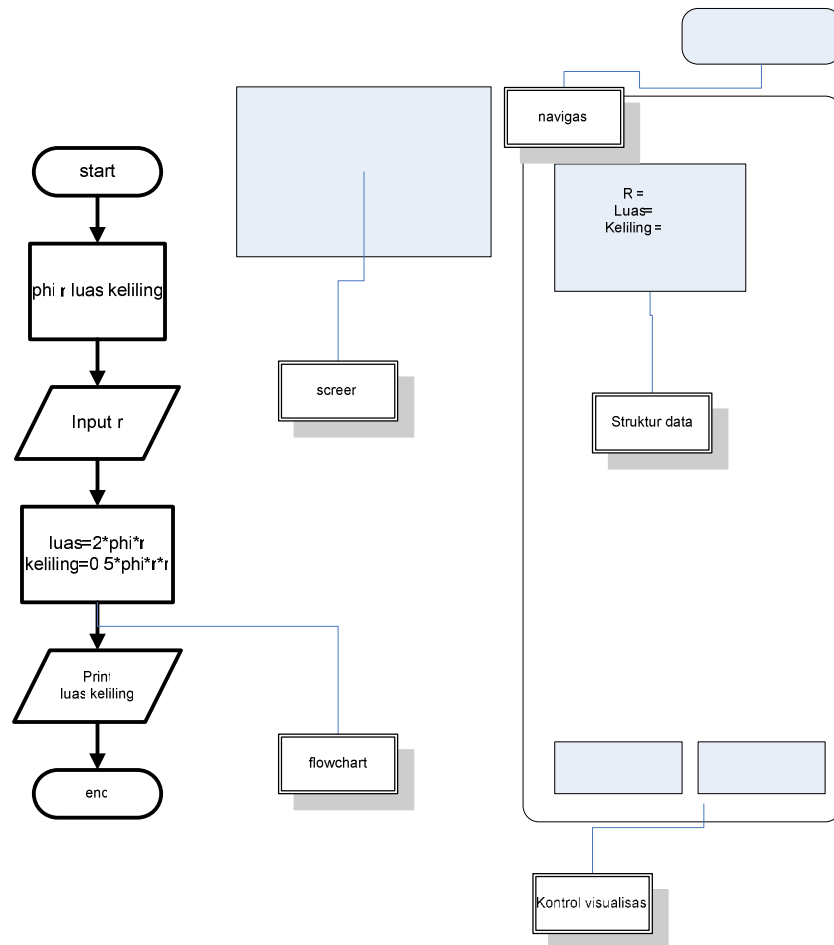
User interface dari Menu visualisasi adalah sebagai berikut :

LOGIKA DAN ALGORITMA



Gambar 3.3 desain menu visualisasi logika algoritma

1.12.3 Perancangan User Interface Visualisasi



Gambar 3.4 desain GUI visualisasi logika algoritma

1.12.4 Perancangan Hardware dan Software untuk Server dan Client

Hardware dan Software yang diperlukan untuk visualisasi ini mempunyai karakteristik sebagai berikut :

Perangkat Keras (server) :

- AMD Sempron 1.6 GHz
- DDR 256 MB
- 40 GB HDD
- Graphic Card 32 MB
- Resolution 1024x768
- Cd-ROM
- Mouse
- Keyboard

Perangkat Keras (client) :

- AMD Sempron 1.6 GHz
- DDR 256 MB
- 40 GB HDD
- Graphic Card 32 MB
- Resolution 1024x768
- Printer
- Mouse
- Keyboard

Perangkat Lunak (client) :

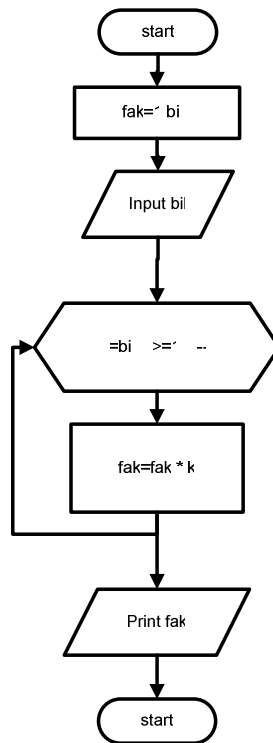
- Apache 2.52
- Antivirus

Perangkat Lunak (client) :

- Browser (Mozilla Firefox 2.0 recommended)
- Plug-in Flash Player v 9.0 untuk mozilla firefox

1.13 ANALISA KEBUTUHAN VISUALISASI

Dalam matematika, faktorial dari bilangan asli n adalah hasil perkalian antara bilangan bulat positif yang kurang dari atau sama dengan n . Faktorial ditulis sebagai $n!$ dan disebut n faktorial. Dalam menghitung faktorial terdapat pengulangan sampai nilai $n=1$. Algoritma bilangan faktorial disajikan dalam flowchart dibawah ini :



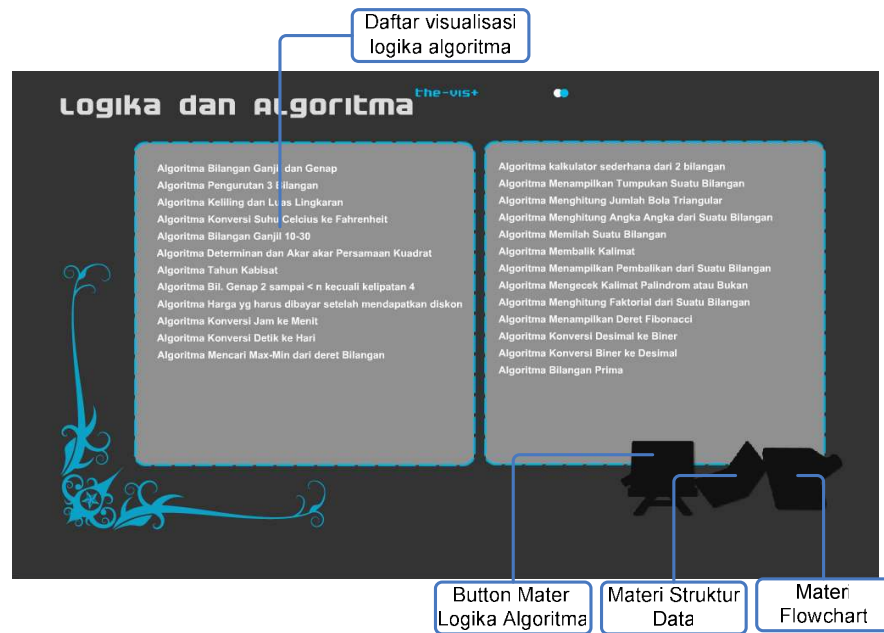
Gambar 3.5 Flowchart Algoritma Bilangan Faktorial

Kebutuhan sistem dari visualisasi dari Algoritma Bilangan Faktorial adalah :

1. Visualisasi menampilkan definisi umum dari algoritma menghitung n faktorial
2. Visualisasi menampilkan flowchart dari algoritma menghitung n Faktorial
3. Setiap bagian dari flowchart akan berubah warna menjadi merah jika dituju.
4. User dapat memberikan input bil untuk divisualisasikan sebagai input dari flowchart
5. Visualisasi hanya dapat menerima masukan berupa angka saja, selain angka, flowchart tidak dapat berjalan ke state selanjutnya.
6. Visualisasi hanya menerima bilangan bulat positif saja, karena dalam algoritma faktorial dijelaskan bahwa faktorial adalah *hasil perkalian antara bilangan bulat positif yang kurang dari atau sama dengan n*.
7. Output dari visualisasi adalah hasil dari n faktorial atau $F(n)$.
8. Visualisasi bisa mencetak tampilan lewat printer

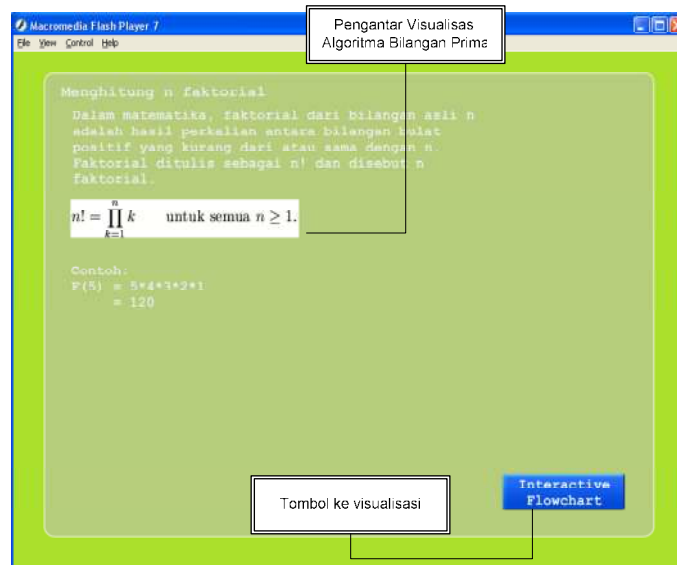
1.14 IMPLEMENTASI

Pada visualisasi ini terdapat beberapa movie, diantaranya adalah :



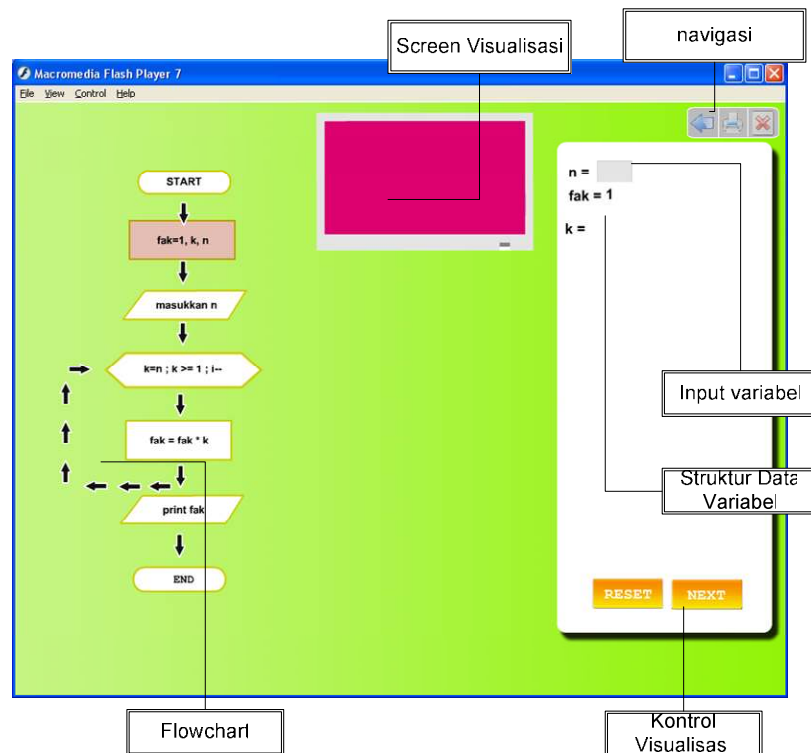
Gambar 3.6 Menu visualisasi logika algoritma

Pada menu movie terdapat daftar 25 algoritma yang akan divisualisasikan. Setiap visualisasi disimpan dalam file html dan untuk memanggilnya akan muncul lewat pop-up. Dibawah ini GUI dari Algoritma Bilangan Faktorial .



Gambar 3.7 GUI dari Algoritma Bilangan Faktorial

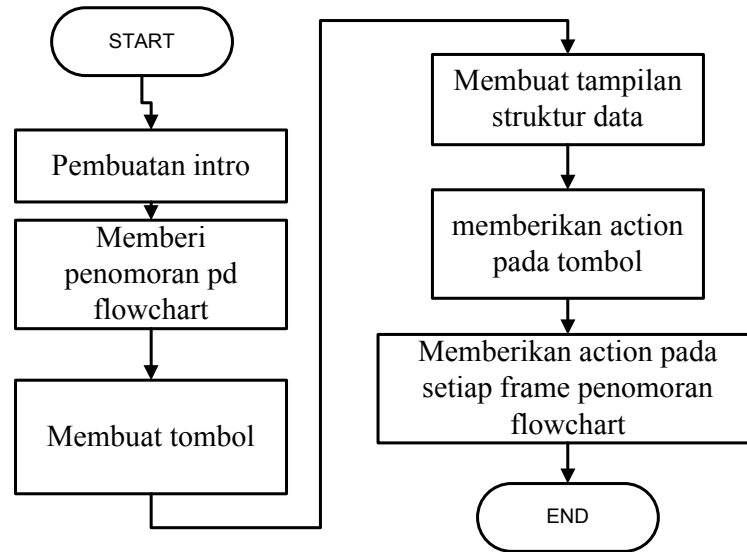
Dibawah ini adalah GUI dari visualisasi Logika Algoritma:



Gambar 3.8 Visualisasi Algoritma Bilangan Faktorial

Dalam visualisasi movie terdapat beberapa tombol navigasi yaitu untuk menutup pop-up, untuk print screen dan untuk kembali ke scene sebelumnya. Selain itu terdapat bagian looping screen yang akan memvisualisasikan pada looping ke berapa suatu algoritma dijalankan.

Dalam sub bab dibawah ini dijelaskan langkah langkah coding dalam membuat visualisasi Algoritma Bilangan faktorial. Dalam fase coding terdapat langkah langkah sebagai berikut :



Gambar 3.9 Diagram Alir coding visualisasi

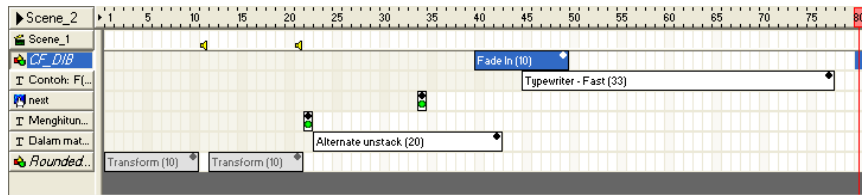
1.14.1 Membuat Intro Visualisasi Algoritma Menghitung n Faktorial

Intro visualisasi berisikan tentang animasi dari definisi faktorial dan contoh faktorial. User interface dari intro visualisasi adalah sebagai berikut:



Gambar 3.10 User interface dari intro visualisasi

Jenis effect yang diterapkan dalam intro adalah Alternate Stack dan Typewriter.



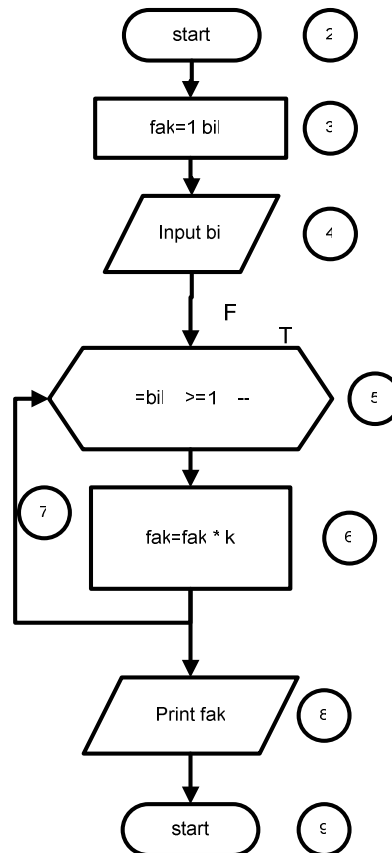
Gambar 3.11 Jenis effect pada intro visualisasi

Untuk button Interactive Flowchart mempunyai script action sebagai berikut :

```
on (rollOver) {
    playSound("Beep2.wav");
}
on (release) {
    gotoSceneAndStop("Scene_2",1);
}
```

1.14.2 Memberi penomoran pada flowchart

Flowchart yang sudah mempunyai penomoran seperti gambar dibawah ini :



Gambar 3.12 Flowchart yang sudah memiliki penomoran

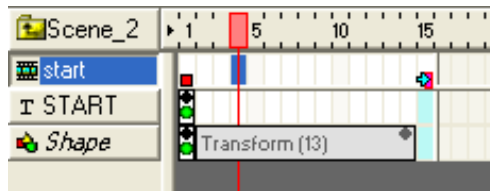
1.14.3 Membuat sprite untuk tiap chart

Setiap chart dari flowchart akan kita buat ke dalam sprite tersendiri. Sebagai contoh untuk chart “start” adalah sebagai berikut :



Gambar 3.13 Sprite untuk chart “start”

Sprite start terdiri dari gambar dan text “START”. Pada timeline sprite tersebut mempunyai effect “Transform” yang akan merubah warna menjadi merah apabila sprite dijalankan.



Gambar 3.14 Efek transform pada shape

Pada saat frame 1, diberi action `Stop()` supaya sprite akan berhenti pada saat frame pertama dan tidak akan melangkah ke frame selanjutnya selama sprite tersebut kita panggil.

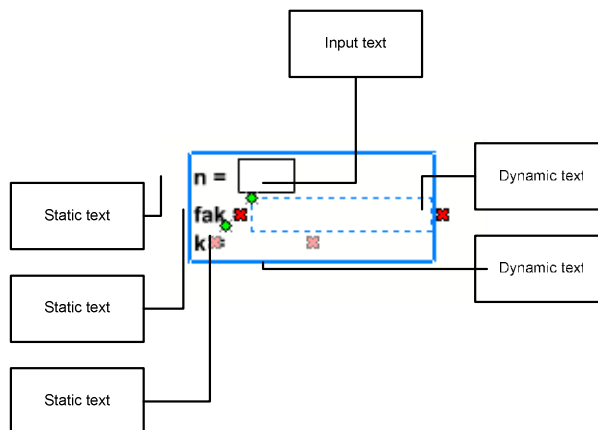
```
onFrame (1) {
    stop();
}

onFrame (15) {
    gotoAndPlay(2);}
```

Sedangkan untuk frame 15 diberikan action `gotoAndPlay(2)` agar sprite akan mengulangi frame 2-15 . Semua chart dibuat dengan cara yang sama dengan chart sprite yaitu mempunyai panjang 15 frame.

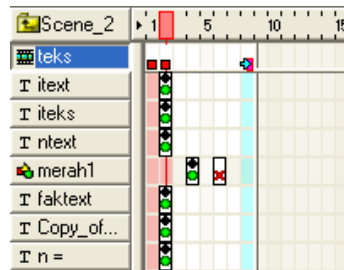
1.14.4 Membuat tampilan struktur data

Struktur data yang dibutuhkan adalah variabel n , fak, dan k . Sebuah input dari user berupa n , fak sebagai variabel penampung hasil faktorial dan k adalah jumlah looping.



Gambar 3.15 Struktur data bilangan faktorial

Tampilan struktur data dibuat pada sebuah sprite dengan nama "teks". Pada frame 1, semua object tidak ditampilkan. Object object tersebut ditampilkan pada frame 2.



Gambar 3.16 Objek objek dalam scene 2

Untuk frame 4 dan 6 diberikan effect "place" dan "remove" pada object "merah1" sehingga apabila sprite dijalankan, kotak merah akan berkedip. Pada saat frame 1, diberi action Stop() supaya sprite akan berhenti pada saat frame pertama dan tidak akan melangkah ke frame selanjutnya selama sprite tersebut kita panggil.

```
onFrame (1) {
    stop();
}
onFrame (8) {
    gotoAndPlay(3);}
```

Sedangkan untuk frame 8 diberikan action gotoAndPlay(3) agar sprite akan mengulangi frame 3-8 .

1.14.5 Membuat tombol dan memberikan action

Tombol yang akan dibuat adalah tombol untuk mengatur jalannya animasi yaitu tombol start, next dan reset. Tombol tersebut mempunyai action tersendiri, seperti tombol start dan next yang berfungsi untuk meloncat ke frame tujuan. Sedangkan tombol reset berfungsi untuk mereset scene 2 sehingga posisi playhead kembali ke frame 1.

Script dari tombol start dan next adalah seperti dibawah ini:

```
on (press) {
    gotoSceneAndStop("Scene_2", tujuan);
}
```

Sedangkan script untuk tombol reset adalah :

```
on (press) {
```

```

_root.gotoAndStop(1);
gotoSceneAndStop("Scene_2",1);
}

```

1.14.6 Memberi action pada setiap penomoran flowchart

Pada frame 1 adalah kondisi awal pada saat scene 1 berhenti. Script pada frame 1 adalah :

```

onFrame (1) {
    tujuan=2;
    stop();}

```

Pada frame 2 atau pada kondisi memulai visualisasi(start), action yang diberikan adalah memanggil sprite “start” dan memberikan nilai variable tujuan = 3. Script pada frame 2 adalah :

```

onFrame (2) {
    tujuan=3;
    start.gotoAndPlay(1);
}

```

Pada frame 3 yaitu pada deklarasi variabel, action yang diberikan adalah menghentikan sprite “start” dan menjalankan sprite “deklarasi”.selain itu variabel tujuan diberi nilai 4. Pada frame 3 terdapat deklarasi variabel.

```

onFrame (3) {
    tujuan=4;
    start.gotoAndStop(1);
    deklarasi.gotoAndPlay(1);
    teks.gotoAndStop(2);
    var fakvar,k,n;
    fakvar=1;
    teks.faktext.text=fakvar;
}

```

Pada frame 4 sprite “inputn” dijalankan sedangkan sprite sebelumnya yaitu sprite “deklarasi” dihentikan.

```

onFrame (4) {
    tujuan=5;
    deklarasi.gotoAndStop(1);
    teks.gotoAndPlay(3);
    inputn.gotoAndPlay(1);
}

```

Pada frame 5 seperti langkah sebelumnya, sprite sebelumnya dihentikan dan input text dari user dimasukkan kedalam variable n.

```
onFrame (5) {n=teks.ntext.text;
  if (validate(n)==false) {
    prevFrameAndStop();
  }else {
    teks.gotoAndStop(2);
    inputn.gotoAndStop(1);
    tujuan=6;
    angka(n);
    looping.gotoAndPlay(1);
    k=n;
    obyek2=eval("loopsprite.lup"+add k);
    obyek2.gotoAndPlay(2);
  }
}
```

Dalam frame 5 input dari user divalidasi supaya input yang masuk hanya berupa angka, yaitu menggunakan fungsi validate().

```
function validate(x) {
  var valid=true;
  var z=x;
  if (z.length >0) {
    for (i = 0; i < z.length; i++) {
      if (z.charCodeAt(i) < 48 || z.charCodeAt(i) > 57) {
        valid = false;
        break;
      }
    }
  }
  else {valid = false;}
  return valid;}

```

Dalam frame 5 memasuki kondisi looping. Untuk menampilkan jumlah looping, kita membuat fungsi yang menduplikasi sprite yang sudah kita buat dan memberi nilai sesuai dengan jumlah looping. Fungsi tersebut adalah fungsi angka(). Dalam fungsi angka, sprite yang digandakan adalah sprite “loopsprite.lup”. parameter fungsi angka adalah n atau jumlah looping.

```
function angka(n) {
  for (i=1; i<=n; i++) {
    loopsprite.lup.duplicateSprite("lup"+i,i);
    xpos=loopsprite.lup._X+(i*25);
    d=eval("loopsprite.lup"+add i);
    d.lup.text=i;
  }
}
```

```

        d._X=xpos;
    }
    loopsprite.lup.unloadMovie();
}

```

Pada frame 6 perhitungan faktorial dilakukan. Script lebih jelasnya seperti dibawah ini.

```

onFrame (6) {
    if (k<n) {
        r=k+1;
        obyek3=eval("loopsprite.lup"add r);
        obyek3.gotoAndStop(1);
    }
    obyek2=eval("loopsprite.lup"add k);
    obyek2.gotoAndPlay(2);
    looping.gotoAndStop(1);
    faktorial.gotoAndPlay(1);
    fakvar=fakvar*k;
    teks.faktext.text=fakvar;
    k--;
    if (k>=1) {
        tujuan=7;
    } else {
        tujuan=8;
    }
}}

```

Pada frame 7 script lebih jelasnya sebagai berikut :

```

onFrame (7) {
    faktorial.gotoAndStop(1);
    printfak.gotoAndStop(1);
    looping.gotoAndPlay(16);
    tujuan=6;    }

```

Pada frame 8 hasil dari perhitungan factorial ditampilkan pada screen. Untuk script lebih jelasnya sebagai berikut:

```

onFrame (8) {
    d=k+1;
    obyek2=eval("loopsprite.lup"add d);
    obyek2.gotoAndStop(1);
    faktorial.gotoAndStop(1);
    printfak.gotoAndPlay(1);
}

```

```
        screen.gotoAndPlay(1);  
        screen.print.text=fakvar;  
        tujuan=9;  
    }
```

Pada frame 9, adalah chart terakhir yaitu End. Sprite yang dijalankan adalah “end” dan menghentikan sprite sebelumnya yaitu “printfak” . Script lebih jelasnya seperti dibawah ini.

```
onFrame (9) {  
    screen.gotoAndPlay(16);  
    printfak.gotoAndStop(1);  
    end.gotoAndPlay(1);  
    tujuan=10;  
}  
onFrame (10) {  
    _root.gotoAndStop(1);  
    gotoSceneAndStop("Scene_2",1);  
}
```

Frame 9 bukan frame yang terakhir melainkan masih ada frame untuk menghentikan semua movie sehingga visualisasi kembali pada kondisi awal yaitu kembali ke frame 1.

BAB IV

PENGUJIAN PROGRAM DAN ANALISA

Uji coba dilakukan pada semua visualisasi yang telah dibuat dengan membandingkan dengan cara menghitung manual sesuai teori yang ada.

1.15 Pengujian

Setiap pengujian dihitung jumlah prosentase perbedaan antara perhitungan manual secara teori dengan apabila menggunakan alat bantu ini. Tiap tiap visualisasi diuji dalam sub-bab dibawah ini.

Pengujian Algoritma Bilangan Ganjil dan Genap

Perhitungan manual :

Input : 5

$5/2 = 4$ sisa 1

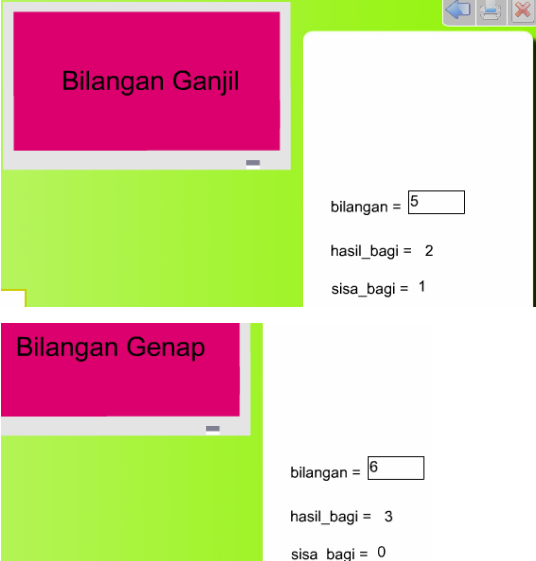
Apa bila sisa=1 maka bilangan itu adalah bilangan ganjil

Input : 6

$6/2 = 3$ sisa 0

Apa bila sisa = 0 maka bilangan itu adalah bilangan genap

Dengan Alat bantu :



The image shows two screenshots of a Java Swing application. The top screenshot is titled "Bilangan Ganjil" and displays the following text: "bilangan = 5", "hasil_bagi = 2", and "sisa_bagi = 1". The bottom screenshot is titled "Bilangan Genap" and displays the following text: "bilangan = 6", "hasil_bagi = 3", and "sisa_bagi = 0". Both screenshots show a simple window with a title bar and a single text area for each input.

Pengujian Algoritma Mengurutkan 3 Bilangan

Perhitungan Manual :

Input :

Bil[1]=22, bil[2]=5, dan bil[3]=1

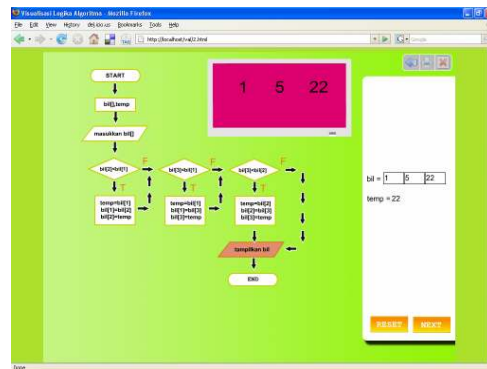
Apabila bil[2] < bil[1], maka temp=22 , bil[1]=5 , bil[2]=22

Apabila bil[3] < bil[1], maka temp=5 , bil[1]=1 , bil[3]=5

Apabila bil[3] < bil[2], maka temp=22 , bil[2]=5 , bil[3]=22

Output bil[1]= 1, bil[2]= 5 , bil[3]=22

Dengan alat bantu :



Pengujian Algoritma Menghitung Luas dan Keliling Lingkaran

Perhitungan Manual :

Input : r=123

Luas Lingkaran = $\pi \cdot r \cdot r$

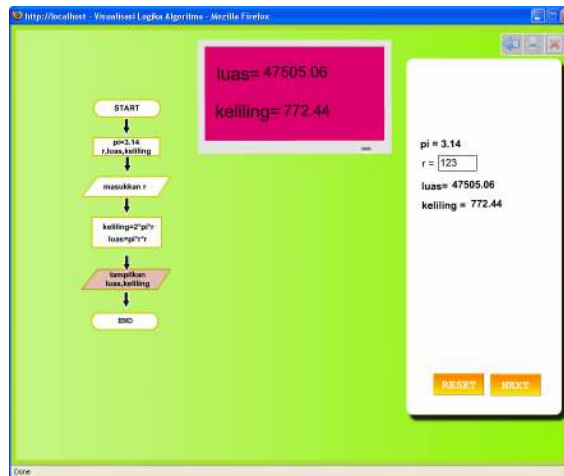
$$= 3.14 \cdot 123 \cdot 123$$

$$= 47505.06$$

Keliling Lingkaran = $2 \cdot \pi \cdot r$

$$= 2 \cdot 3.14 \cdot 123 = 772.44$$

Dengan alat bantu :



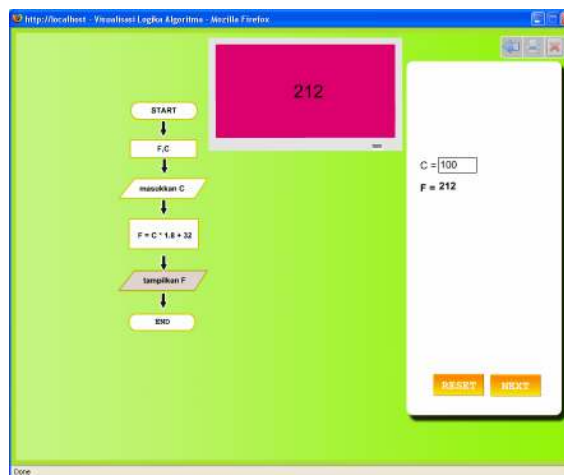
Pengujian Algoritma Konversi Suhu Celcius Ke Fahrenheit

Perhitungan Manual :

Input : C = 100

$$\begin{aligned}\text{Suhu dalam Fahrenheit} &= (C * 1.8) + 32 \\ &= (100 * 1.8) + 32 \\ &= 180 + 32 = 212\end{aligned}$$

Dengan alat bantu :



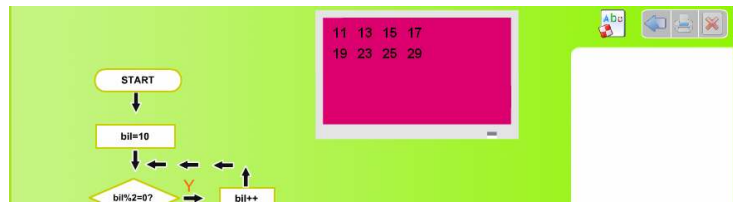
Pengujian Algoritma Bilangan Ganjil antara 10-30

Perhitungan Manual :

Input : -

Output = 11 ,13 ,15 ,17 ,19, 23, 25, 29

Dengan alat bantu :



Pengujian Algoritma Determinan dan Akar Persamaan kuadrat

Perhitungan Manual :

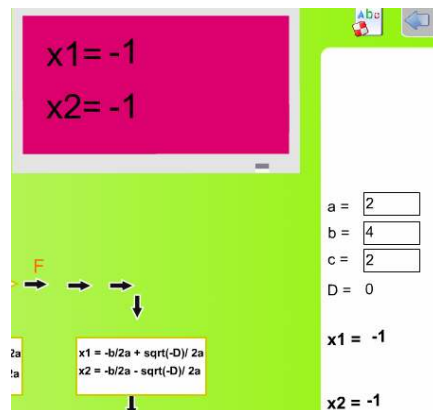
Input : a= 2, b=4, c=2

$$D = b^2 - 4 * a * c = 4^2 - 4 * 2 * 2 = 0$$

$$x1 = -b / 2a = -4 / 2 * 2 = -1$$

$$x2 = x1 = -1$$

Dengan alat bantu :



Pengujian Algoritma Tahun Kabisat

Perhitungan Manual :

Input : tahun =2005

Tahun 2005 bukan tahun kabisat karena tidak habis(mempunyai sisa yaitu 1 apabila dibagi dengan 4)

Dengan alat bantu :



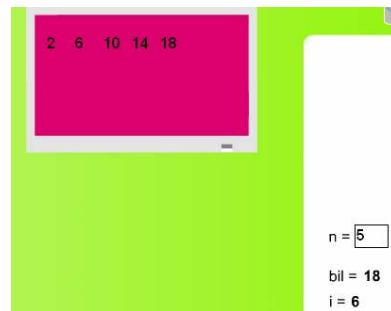
Pengujian Algoritma n bilangan genap kecuali kelipatan 4

Perhitungan Manual :

Input : 5

5 bilangan genap dimulai dua kecuali bilangan itu kelipatan 4 adalah 2,6,10,14,18

Dengan alat bantu :



Pengujian Algoritma diskon

Perhitungan Manual :

Input : jumlah barang =2

barang[1]= sapi

harga[1]= 1000.000

barang[2]= kambing

harga[2]= 1000.000

jumlah total = 2000.000

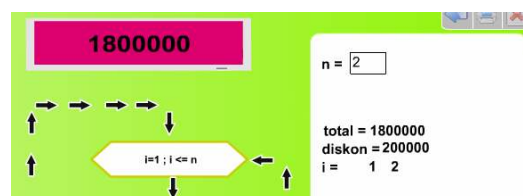
apabila lebih dari 1500.000 maka mendapatkan diskon 10%

Jumlah total setelah mendapatkan diskon = $2000.000 - 10\% \times 2000.000$

= $2000.000 - 200.000$

= 1800.000

Dengan alat bantu :



Pengujian Algoritma konversi jam ke menit

Perhitungan Manual :

Input : jam =4, menit =56

$$\begin{aligned}\text{Menit} &= (\text{jam} \times 60) + \text{menit} \\ &= (4 \times 60) + 56 = 296\end{aligned}$$

Dengan alat bantu :



Pengujian Algoritma konversi detik ke hari

Perhitungan Manual :

Input : bil = 110000

$$\text{detik} = \text{bil} \% 60 = 110000 \% 60 = 20$$

$$\text{bil} = \text{bil} / 60 = 110000 / 60 = 1833$$

$$\text{menit} = \text{bil} \% 60 = 1833 \% 60 = 33$$

$$\text{bil} = \text{bil} / 60 = 1833 / 60 = 30$$

$$\text{jam} = \text{bil} \% 24 = 30 \% 24 = 6$$

$$\text{hari} = \text{bil} / 24 = 30 / 24 = 1$$

Dengan alat bantu :



Pengujian Algoritma max min dari deret bilangan

Perhitungan Manual :

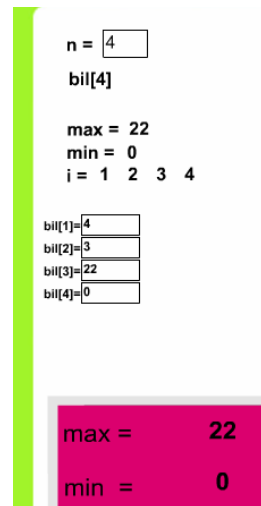
Input : jumlah deret = 4

deret = 4,3,22,0

max = 22

min = 0

Dengan alat bantu :



Pengujian Algoritma kalkulator sederhana 2 bilangan

Perhitungan Manual :

Input : bil1 = 15, bil2 = 2, op = "/"

hasil = bil1 / bil2 = 15/2 = 7.5

Dengan alat bantu :



Pengujian Algoritma menampilkan tumpukan bilangan

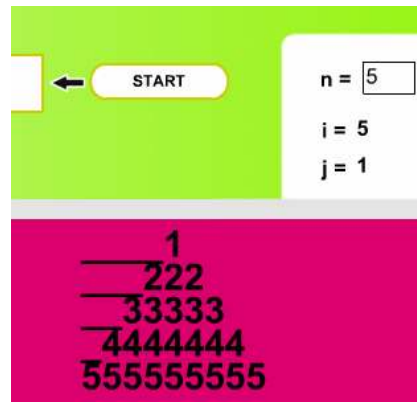
Perhitungan Manual :

Input : n = 5

Output

1
222
333333
44444444
555555555

Dengan alat bantu :



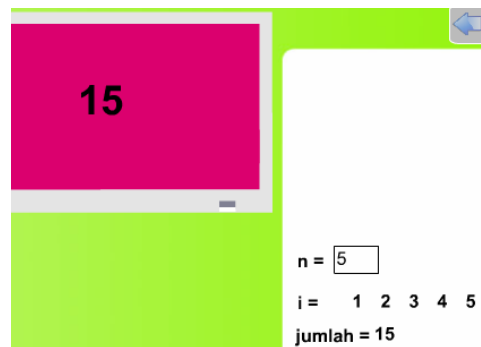
Pengujian Algoritma jumlah bola triangular

Perhitungan Manual :

Input : $n=5$

Jumlah bola triangular jika bagian bawah = 5 yaitu $5+4+3+2+1$
= 15

Dengan alat bantu :



Pengujian Algoritma menghitung angka angka dari bilangan

Perhitungan Manual :

Input : $bil = 3456$

Jumlah angka angka adalah $3+4+5+6 = 18$

Dengan alat bantu :



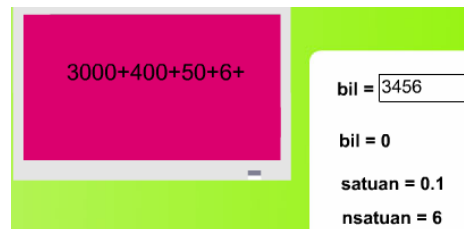
Pengujian Algoritma memilah suatu bilangan

Perhitungan Manual :

Input : bil = 3456

Output : $3000 + 400 + 50 + 6$

Dengan alat bantu :



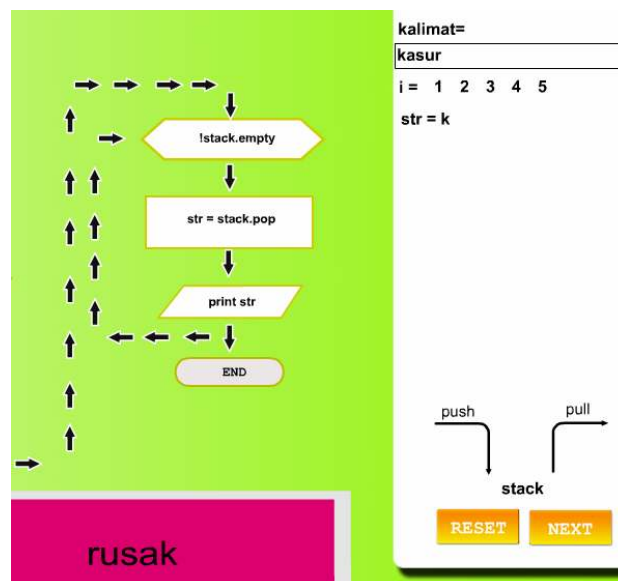
Pengujian Algoritma membalik kalimat

Perhitungan Manual :

Input : kalimat = kasur

Output : rusak

Dengan alat bantu :



Pengujian Algoritma membalik bilangan

Perhitungan Manual :

Input : bil = 234

Output : 432

Dengan alat bantu :



bil =

nilai = 4

satuan = 1000

hasil = 432

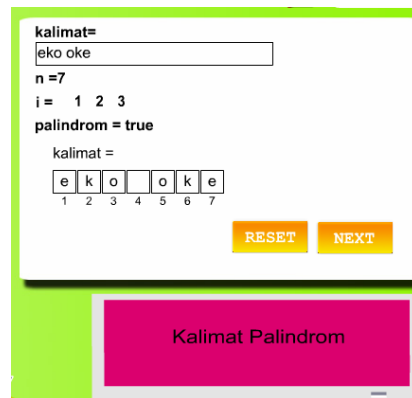
Pengujian Algoritma kalimat palindrom

Perhitungan Manual :

Input : kalimat = "eko oke"

Output : "kalimat palindrom"

Dengan alat bantu :



kalimat=

n = 7

i = 1 2 3

palindrom = true

kalimat =

e	k	o	o	k	e	
1	2	3	4	5	6	7

Kalimat Palindrom

Pengujian Algoritma bilangan faktorial

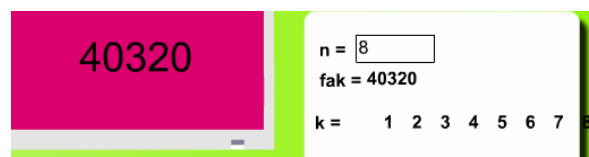
Perhitungan Manual :

Input : bil = 8

Output : bilangan faktorial dari 8 adalah

$$F(8) = 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 40320$$

Dengan alat bantu :



n =

fak = 40320

k = 1 2 3 4 5 6 7 8

Pengujian Algoritma deret fibbonaci

Perhitungan Manual :

Input : bil = 5

Output : 5 deret fibonacci adalah

fib(0) = 0

fib(1) = 1

fib(2) = fib(0) + fib(1) = 1

fib(3) = fib(1) + fib(2) = 2

fib(4) = fib(3) + fib(2) = 3

Dengan alat bantu :



Pengujian Algoritma Desimal ke Biner

Perhitungan Manual :

Input bilangan = 45

$45 / 2 = 22$ sisa 1

$22 / 2 = 11$ sisa 0

$11 / 2 = 5$ sisa 1

$5 / 2 = 2$ sisa 1

$2 / 2 = 1$ sisa 0

$1 / 2 = 0$ sisa 1

Jadi bilangan biner dari 45 adalah 101101

Dengan alat bantu :



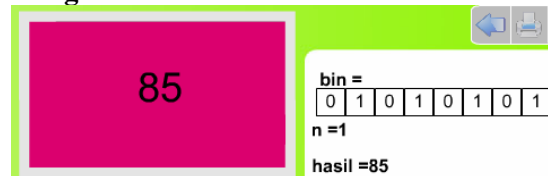
Pengujian Algoritma Biner ke Desimal

Perhitungan Manual :

Contoh bil = 1010101

1	0	1	0	1	0	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	0	16	0	4	0	1
$\Rightarrow 64+0+16+0+4+0+1 = 85$						

Dengan alat bantu :



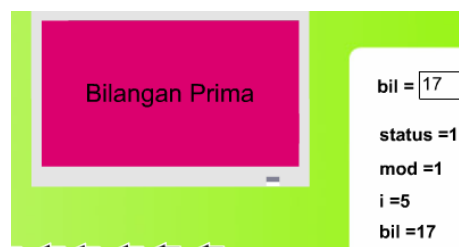
Pengujian Algoritma Bilangan Prima

Perhitungan Manual :

Input = 17

Output = "bilangan Prima" karena $17\%17 = 0$ dan $17\%1 = 0$

Dengan Alat Bantu :



1.16 ANALISA DAN HASIL UJI COBA

Dari hasil pengujian tersebut dapat dihitung prosentase kesalahan dari visualisasi algoritma bilangan faktorial.

$$\text{Prosentase pengujian sistem} = \frac{\text{jumlah_output_yang_sama}}{\text{jumlah_pengujian}} \times 100\%$$

$$= \frac{25}{25} \times 100\%$$

$$= 100 \%$$

Algoritma algoritma yang divisualisasikan sudah sesuai dengan syarat algoritma yang dikemukakan oleh Donald E. Knuth yaitu:

6. *Finiteness*, Algoritma berakhir(terminate) setelah melakukan sejumlah langkah proses dan menghasilkan output.
7. *Definiteness*, Setiap langkah algoritma didefinisikan dengan tepat dan tidak menimbulkan makna ambigu.
8. *Input*, Visualisasi Algoritma mempunyai input data untuk diolah. Apabila input data tidak diisi maka visualisasi tidak akan melanjutkan ke proses selanjutnya.
9. *Output*, Visualisasi Algoritma memberikan output setelah mengerjakan proses .
10. *Effectiveness*, langkah langkah algoritma dikerjakan dalam waktu yang wajar yakni output keluar langsung setelah melakukan proses.

BAB V

KESIMPULAN DAN SARAN

1.17 KESIMPULAN

Dari pembahasan pada bab-bab sebelumnya dapat ditarik kesimpulan :

1. Langkah langkah dalam pembuatan perangkat lunak alat bantu logika dan algoritma adalah Analisa , Desain, Implementasi dan Pengujian.
2. Yang mempengaruhi baik buruknya visualisasi adalah resolusi visualisasi, struktur data yang divisualisasikan, interaktifitas dari visualisasi, dan *Computational infrastructure*.
3. Dalam menyusun algoritma yang baik dan benar harus memenuhi Finiteness(Algoritma harus berakhir atau terminate), Definiteness(algoritma harus didefinisikan dengan tepat), Input (algoritma harus mempunyai input), Output(alagitma harus mempunyai Output), Effectiveness (langkah langkah algoritma dikerjakan dalam waktu yang wajar).

1.18 SARAN

Untuk pengembangan sistem yang telah dibuat, maka diperlukan beberapa perbaikan diantaranya adalah :

1. Perlu dibuat visualisasi versi offline sehingga user bisa memakai visualisasi pada saat menggunakan komputer yang tidak terhubung pada jaringan.
2. Visualisasi dimunculkan lewat pop-up, sedangkan apabila browser men-*disabled* pop-up, maka visualisasi tidak bisa muncul. Jadi untuk kedepan perlu dibuat visualisasi yang muncul tanpa lewat pop-up.

DAFTAR PUSTAKA

- [1] Kristanto, Andri. 2003. *Algoritma dan Pemrograman dengan C++*. Yogyakarta : Graha Ilmu.
- [2] Munir, Rinaldi. 1999. *Algoritma dan Pemrograman dalam bahasa Pascal dan C*. Bandung : Informatika.
- [3] Ngoen, Thompson Susabda. 2004. *Pengantar Algoritma dengan bahasa C*. Jakarta : Penerbit Salemba Teknik.
- [4] Anonim. 2002. *About This Particular Macintosh Magazine*.
- [5] Barakbah, Ali Ridho. 2005. *Logika Dan Algoritma*. Surabaya : PENS-ITS.
- [6] Williams, Laurie. 2004. *Black Box testing*.
- [7] Pemodelan waterfall. www.elektroindonesia.com (didownload 23-3-2007)
- [8] Knuth, Donald E. 1997. *The Art of Computer Programming*. Addison Wesley.
- [9] Hansen, Charles D. 2005. *The Visualization Handbook*.

RIWAYAT HIDUP PENULIS

Nama : Galuh Prasetyawan

NRP : 74 03 040 009

Tempat/tgl lahir : Malang , 2 Juni 1985

Alamat : Sanan bawah 66 Malang 65122

e-mail : techz.boy@gmail.com

blog : <http://prasetyawan.blogspot.com>



**Riwayat pendidikan : Tk. Muslimat NU 22 Malang (-)
SDN. Bunul Rejo V Malang (1993-1999)
SLTPN 3 Malang (1999-2001)
SMUN 8 Malang (2001-2003)
Joint Program VEDC BA-Malang (2003-2007)**