

**Pengaruh Model Pembelajaran Menggunakan *IndoBlockly* (Bahasa Pemrograman Visual Block) terhadap Pemahaman Mahasiswa pada Matakuliah Pemrograman Terstruktur.**

**Studi Kasus Mahasiswa Semester I Angkatan 2012/2013 Teknik Informatika  
UIN Sunan Kalijaga Yogyakarta**



**SKRIPSI**

Diajukan Kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri Sunan Kalijaga  
Untuk Memenuhi Sebagian Syarat Memperoleh Gelar Sarjana  
Strata Satu Teknik Informatika

**Disusun oleh:**

**Rischan Mafrur  
NIM. 09650007**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA  
YOGYAKARTA  
2012**

Pengesahan scan

Persetujuan skripsi

Keaslian Skripsi scan

## KATA PENGANTAR

*Alhamdulillah* rabbi' alamin. Puji syukur bagi Allah Subhanahu wa Ta'ala yang telah melimpahkan rahmat-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi dengan judul

Pengaruh Model Pembelajaran Menggunakan *IndoBlockly* (Bahasa Pemrograman Visual Block) terhadap Pemahaman Mahasiswa pada Matakuliah Pemrograman Terstruktur.

Studi Kasus Mahasiswa Semester I Angkatan 2012/2013 Teknik Informatika UIN Sunan Kalijaga Yogyakarta

dengan lancar dan tanpa suatu halangan apapun. Sholawat dan Salam senantiasa penulis haturkan kepada junjungan nabi agung, Muhammad Shollallahu'alaihi wa Sallam.

Selanjutnya penulis mengucapkan terima kasih kepada:

1. Ibunda tercinta dan seluruh anggota keluarga tersayang yang senantiasa mendo'akan dan memberikan support.
2. Prof. Drs. H. Akh. Minhaji, M.A., Ph.D. selaku Dekan Fakultas Sains & Teknologi UIN Sunan Kalijaga.
3. Bapak Agus Mulyanto, M.Kom, selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga.
4. Bapak Agung Fatwanto, S.Si., M.Kom., Ph.D, selaku Pembimbing yang dengan kesabarannya telah membimbing selama ini.
5. Bapak Sumarsono, S.T., M.Kom., yang sudah memberikan ijin penelitian di kelas praktikum Pemrograman Terstruktur Teknik Informatika 2012/2013.
6. Bapak Romi Satrio Wahono yang sudah memberikan berbagai pencerahan mengenai penelitian ini.
7. Mas Adit laporan pendidikan matematika yang sudah memberikan banyak ilmu mengenai statistik.
8. Para Dosen Program Studi Teknik Informatika yang telah memberi bekal ilmu pengetahuan kepada penulis, semoga ilmunya menjadi amal jariyah di dunia hingga akhirat.
9. Angga Maulana, Damar Mustiko Aji, Rosan Qodirin, Ahmad Syaiful, dan Agus Hidayatullah sebagai pengembang *IndoBlockly*.
10. Teman-teman Program Studi Teknik Informatika, khususnya angkatan 2009 yang telah banyak memberi dukungan.

11. Semua pengurus Yayasan Masjid Prayan Raya beserta teman-teman takmir Masjid Prayan Raya yang banyak memberi dukungan.

Penulis merasa masih banyak sekali kekurangan dan kelemahan dalam penelitian ini, oleh karena itu segala kritik dan saran senantiasa penulis harapkan dari para pembaca. Akhir kata, semoga penelitian ini dapat menjadi panduan serta referensi yang sangat berguna bagi pembaca dan dapat dimanfaatkan sebaik-baiknya.

Yogyakarta,  
Yang Menyatakan

Rischan Mafrur  
NIM. 09650007

## HALAMAN PERSEMBAHAN

*Kupersembahkan untuk :*

- ❖ Sembah sujudku kepada Allah Subhanahu wa Ta'ala yang senantiasa melimpahkan rahmat dan hidayah-Nya.
- ❖ Sholawat dan Salam kepada junjungan nabi besar Muhammad Shollallahu'alaihi wa Sallam.
- ❖ Ibuku, Ibuku, Ibuku dan ayahku tercinta yang tak pernah henti-hentinya berjuang demi aku. Semoga anakmu ini bisa membahagiakan ibu dan bapak, dan mohon doanya wahai ibu dan bapakku semoga anakmu ini bisa memberikan manfaat bagi ibu bapak, keluarga, masyarakat, bangsa dan umat islam ini.
- ❖ Adikku tersayang Anjani, jangan nakal yah, belajar yang tekun supaya besok bisa lebih baik dari masmu ini :-D.
- ❖ Bapak Agus Mulyanto yang selama keberadaan penulis dalam menuntut ilmu di Jogja ini sudah seperti ayah kedua bagi penulis dan selaku murrabi bagi penulis juga yang selalu sabar membimbing dan memberikan nasehat-nasehat dan saran yang tak ternilai harganya. Semoga Allah selalu melindungi Pak Agus sekeluarga.
- ❖ Bapak Agung Fatwanto yang telah banyak sekali memberikan ilmu-ilmu dan diskusi yang bisa menambah pengetahuan penulis. Semoga Allah senantiasa memberikan kemudahan dan petunjuk-Nya untuk Pak Agung dan tak lupa semoga dengan bimbingan Allah saya segera mengikuti jejak Pak Agung.
- ❖ Bapak Romi Satrio Wahono, terimakasih atas diskusinya pak dan juga dukungannya terhadap penelitian saya.
- ❖ Bapak Sumarsono terimakasih banyak atas ijinnya untuk melakukan penelitian di kelas praktikum Pemrograman Terstruktur pak, dan

terimakasih banyak atas masukan-masukkannya pak. Semoga Allah melindungi Pak Sumarsono dan keluarga.

- ❖ Bapak Anshori yang walaupun belum pernah mengajar penulis dan hanya ketemu ngobrol sesaat tapi dari motivasi beliau penulis menjadi selalu bersemangat.
- ❖ Ibu Uyun yang selalu asik diajak diskusi mengenai AI ayoo maju AI Indonesia hehe :-D. Semangat terus bu mendidik calon2 peneliti Informatika dari UIN Sunan Kalijaga.
- ❖ Dosen Teknik Informatika: Pak Nurochman terimakasih ilmu JST'nya pak, Pak Aulia doakan biar cepet gak jomblo pak hee :-D, Pak Taufik, Pak Mustakim, Pak Bambang, Pak Didik, Ibu Ulfa, Ibu Ade , terimakasih untuk semua Ilmu yang sudah diajarkan, ilmu ini akan menjadi amal jariyah yang tidak akan pernah putus, Semoga Allah melindungi Bpk ibu Dosen semuanya.
- ❖ Teman-teman terdekatku Fadli (he's no life :-D coding terus ...) dan Krocol/Sholahudin (kembang asem) terimakasih tumpangan kosnya yang hampir setiap hari aku selalu disitu :-D.
- ❖ Teman-teman terdekatku Hafidh "oyo prengas prenges ae fid", pulung aktipis KRPH tetep istiqomah lung, udin "oyo salah ngriting din, saake umimu :-D", Aspar "semoga menjadi ulil albab (orng2 yang berpikir) :-D", Anik "entah sampai sekarang gak tau kamu itu cewe ato cowo nik :-D". Terimakasih teman-teman ayoo ndang nyusul, ndang lulus, wis di usir pak Agus heee :-D.
- ❖ Seluruh teman-teman Teknik Informatika: Kambing, Yosep, Ahdi, Pasa, Estu, Sigit, Oki, Iza, Kusuma, Yanuar, Dimas, Kiki, Ayu, Ratna, Ulin, Delisa, Disa, devi, Amy, Ismi, latip, joko, Lukman dan masih banyak lagi yang tidak bisa kusebutkan satu persatu, "KELUARGA BESAR TIF 09.. KESUKSESAN ADA DI DEPAN KITA.. SEMANGAT!!!!!!".
- ❖ Iostream.in crew (informatics research team), salam srigala berkepala 9.



- ❖ *Mas-mas alumni Teknik Informatika yang menginspirasi, Mas Ganjar, Mas Rifki, Mas Alex, semoga bisa menyusul mas :-D. Mas Fathan yang selalu ngasih nasehat untuk segera nikah dan yang pasti yang sering ngasih siswa untuk private atau proyek juga, Mas Sigit, Mas Sunu, Mas Fendi, Mas Veta, Mas Saiful, Mas Budeng, Warok Ngalek, makasih ilmu-ilmu yang sudah ditularkan mase :-D.*
- ❖ *Teman-teman Laboran Mas Adit laboran PMat yang sudah banyak membantu SPSS, Mas Iqbal, Mas Nawir, Mas Rian, dan Mas Yusuf, Terimakasih banyak mas, sudah sering ngrepoti selama ini.*
- ❖ *Seluruh teman-teman pengembang IndoBlockly : Angga, Damar, Rosan, Syaiful dan Agus. Tetep maju IndoBlockly untuk bangsa Indonesia :-D.*
- ❖ *Bapak Nur Mukhlis selaku Ketua Takmir Masjid Prayan Raya yang sudah memberikan ijin dari semester 1 sampai saat ini untuk tinggal di Masjid Prayan Raya, dan juga seluruh warga Prayan Kulon yang sudah saya anggap sebagai keluarga.*
- ❖ *Teman-teman takmir Masjid Prayan Raya : Gembuskun, hendrik, mas ompol dewo, mas hilmy, teman-teman seperjuangan yang berusaha selalu memakmurkan masjid, maaf yah klo tidur di atas subuh sering krinan :-D #krinan koq bendino.*
- ❖ *Teman-teman KKN yang penuh kenangan: Irul, Zaid, Habib, Wahyu, Sulis, Julida, Asti, Iis, Nurika, Ibu Sekar.*
- ❖ *Teman-teman Imagine : Pak Bos Pak Agung, Mbak Esa, Mas Arul, Mas Arfin, Joko, Rio, Gabriel dsb.*
- ❖ *Yang terakhir adalah seseorang yang ada disana untuk menunggu kedatanganku, semoga Allah menyegerakan dan slalu memberikan jalan yang terbaik bagi kita. Allahu A'lam ..... :-D.*

## **HALAMAN MOTTO**

**#Sabar | #Syukur**

**{Sabar dalam nikmat;} | {syukur dalam musibah;}**

## DAFTAR ISI

Halaman Judul.....	i
Pengesahan Skripsi/Tugas Akhir .....	ii
Surat Persetujuan Skripsi/Tugas Akhir .....	iii
Pernyataan Keaslian Skripsi.....	iv
Kata Pengantar .....	v
Halaman Persembahan .....	vii
Halaman Motto.....	x
Daftar Isi.....	xi
Daftar Tabel .....	xv
Daftar Gambar .....	xvi
Daftar Lampiran .....	xvii
Abstraksi .....	xviii
Abstract .....	xix
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.6 Keaslian Penelitian.....	6
<b>BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....</b>	<b>7</b>
2.1 Tinjauan Pustaka .....	7
2.2 Landasan Teori.....	13

2.2.1 Pengertian Belajar .....	13
2.2.2 Media Pembelajaran .....	14
2.2.3 Proses Pembelajaran.....	15
2.2.3.1 Pengertian Ranah Penilaian Kognitif.....	17
2.2.3.2 Ranah Afektif dan Psikomotorik .....	20
2.2.4 Belajar Pemrograman (Learning Programming) .....	21
2.2.5 Bahasa Pemrograman Visual (Visual Programming Language /VPLs) .....	24
2.2.5.1 Kelebihan dan Kekurangan Bahasa Pemrograman Visual .....	25
2.2.5.1.1 Kelebihan VPLs .....	25
2.2.5.2 Kekurangan VPLs.....	26
2.2.6 IndoBlockly .....	27
2.2.6.1 Konsep IndoBlockly .....	27
2.2.6.2 Design Interface IndoBlockly .....	28
2.2.6.3 Contoh Pembuatan Program Sederhana menggunakan IndoBlockly ...	33
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>36</b>
3.1 Metode Penelitian.....	36
3.2 Populasi dan Sampel .....	36
3.2.1 Populasi .....	37
3.2.2 Sampel .....	37
3.3 Desain Penelitian.....	37
3.4 Alur Penelitian .....	39
3.5 Prosedur Penelitian.....	39
3.5.1 Persiapan Penelitian .....	39
3.5.2 Pelaksanaan Penelitian .....	40
3.5.3 Penyelesaian Penelitian .....	40

3.6 Instrumen Penelitian.....	41
3.6.1 Seperangkat Soal .....	41
3.6.1.1 Menentukan Validitas Soal.....	42
3.6.1.2 Menentukan Reabilitas Soal .....	43
3.6.1.3 Menentukan Daya Pembeda Soal .....	44
3.6.1.4 Menentukan Tingkat Kesukaran Soal.....	45
3.6.2 Hasil Uji Instrumen .....	46
3.6.2.1 Hasil Uji Instrumen Expert.....	47
3.6.2.2 Hasil Uji Instrumen Uji Coba .....	48
3.7 Teknik Pengumpulan Data.....	50
3.8 Teknik Pengolahan Data .....	50
3.8.1 Pengolahan Data Kuantitatif .....	50
3.8.1.1 Analisis Data Tes.....	50
3.8.1.1.1 Analisis Deskriptif .....	51
3.8.1.1.2 Analisis Inferensi .....	52
3.8.1.1.2.1 Uji Normalitas .....	52
3.8.1.1.2.2 Uji Homogenitas Varians .....	53
3.8.1.1.2.3 Uji Perbedaan Dua Rata-rata.....	53
3.8.1.1.2.4 Analisis Korelasi Pretes Postes Kelas Ekperimen.....	53
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>55</b>
4.1 Hasil Penelitian dan Pembahasan.....	55
4.1.1 Analisis Deskriptif Data Hasil Pretes dan Postes.....	55
4.1.2 Analisis Inferensi Data Hasil Pretes dan Postes .....	57
4.1.2.1 Analisis Data Kemampuan Awal Mahasiswa.....	58
4.1.2.1.1 Uji Normalitas Data Pretes .....	61

4.1.2.1.2 Uji Perbedaan Dua Rata-rata Data Pretes .....	62
4.1.2.2 Analisis Data Kemampuan Akhir Mahasiswa .....	63
4.1.2.2.1 Uji Normalitas Data Postes .....	66
4.1.2.2.2 Uji Perbedaan Dua Rata-rata Data postes .....	67
4.1.2.3 Analisis Data Kualitas Peningkatan Kemampuan Mahasiswa .....	68
4.1.2.4 Analisis Data Korelasi Pretes Postes Kelas Eksperimen .....	70
<b>BAB V PENUTUP .....</b>	<b>74</b>
5.1 Kesimpulan .....	74
5.2 Saran.....	74
<b>DAFTAR PUSTAKA .....</b>	<b>75</b>
<b>LAMPIRAN.....</b>	<b>78</b>

## DAFTAR TABEL

Tabel 3.1 Desain Kelompok Eksperimen dan Kontrol Pretes-Posttes .....	38
Tabel 3.2 Klasifikasi Koefisien Validitas .....	42
Tabel 3.3 Klasifikasi Koefisien Reliabilitas.....	44
Tabel 3.4 Klasifikasi Koefisien Daya Pembeda.....	45
Tabel 3.5 Klasifikasi Koefisien Indeks Kesukaran .....	46
Tabel 3.6 Hasil Analisis Instrumen Tes Uji Coba.....	49
Tabel 3.7 Teknik Pengumpulan Data.....	50
Tabel 3.8 Kategori Gain yang dinormalisasi.....	51
Tabel 3.9 Kriteria Penilaian Korelasi.....	54
Tabel 4.1 Statistik Deskriptif Data Hasil Pretes dan Postes.....	56
Tabel 4.2 Data Statistik Skor Pretes Kelas Eksperimen dan Kelas Kontrol.....	58
Tabel 4.3 Hasil Uji Normalitas Data Pretes .....	61
Tabel 4.4 Hasil Uji Mann-Whitney Data Pretes .....	63
Tabel 4.5 Statistika Deskriptif Skor Postes Kelas Eksperimen dan Kelas Kontrol...	64
Tabel 4.6 Hasil Uji Normalitas Data Postes.....	67
Tabel 4.7 Hasil Uji Mann Whitney Data Postes .....	68
Tabel 4.8 Statistik Deskriptif Skor Indeks Gain Tes Kemampuan Mahasiswa Kelas Eksperimen dan Kelas Kontrol .....	69
Tabel 4.9 Daftar Persentase Kualitas Peningkatan Kemampuan Mahasiswa Kelas Eksperimen dan Kelas Kontrol .....	69
Tabel 4.10 Hasil Uji Normalitas Data Pretes Postes kelas eksperimen .....	71
Tabel 4.11 Koefisien Korelasi antara Pretes dan Postes Kelas Eksperimen .....	71
Tabel 4.12 Uji Signifikansi Koefisien Korelasi antara Pretes dan Postes Kelas Eksperimen .....	72

## DAFTAR GAMBAR

Gambar 2.1 Hirarki Piramida Taksonomi Bloom .....	20
Gambar 2.2 Konsep IndoBlockly .....	27
Gambar 2.2 Index awal aplikasi IndoBlockly .....	29
Gambar 2.3 Input output sederhana menggunakan IndoBlockly .....	30
Gambar 2.4 Menu Teks pada IndoBlockly .....	30
Gambar 2.5 Contoh penggunaan variabel pada IndoBlockly .....	31
Gambar 2.6 Contoh array di IndoBlockly .....	31
Gambar 2.7 Penggunaan if pada IndoBlockly .....	32
Gambar 2.8 Penggunaan for pada IndoBlockly .....	32
Gambar 2.9 Program sederhana untuk mencari nilai maksimum dari inputan user menggunakan IndoBlockly .....	33
Gambar 2.10 Aplikasi sederhana perhitungan luas persegi panjang menggunakan IndoBlockly .....	34
Gambar 2.11 Hasil output running aplikasi sederhana perhitungan luas persegi panjang menggunakan IndoBlockly .....	34
Gambar 2.12 Hasil konvert code puzzle IndoBlockly ke source code C .....	35
Gambar 2.13 Hasil output running program dengan C Free .....	35
Gambar 3.1 Alur Penelitian .....	38
Gambar 4.1 Hasil Pretes Kelas Eksperimen dan Kelas Kontrol .....	59
Gambar 4.2 Q-Q Plot pretes kelas eksperimen .....	60
Gambar 4.3 Q-Q Plot pretes kelas kontrol .....	60
Gambar 4.4 Hasil Postes Kelas Eksperimen dan Kelas Kontrol .....	64
Gambar 4.5 Uji Normalitas dengan Q-Q Plot data postes kelas eksperimen .....	65
Gambar 4.6 Uji Normalitas dengan Q-Q plot data postes kelas kontrol .....	65



## DAFTAR LAMPIRAN

Lampiran A Pelaksanaan Penelitian.....	79
Lampiran B Analisis Uji Coba Instrumen.....	80
Lampiran C Soal Pretes dan Postes.....	83
Lampiran D Perolehan Data dan Analisis Deskriptif.....	85
Lampiran E Uji Normalitas Data Pretes.....	87
Lampiran F Uji Beda Rata-rata Pretes .....	88
Lampiran G Uji Normalitas Data Postes.....	89
Lampiran H Uji Beda Rata-rata Postes .....	90
Lampiran I Perhitungan Index Gain.....	91
Lampiran J Analisis Korelasi, Uji Signifikansi Korelasi, dan Determinasi.....	92

## INTISARI

Kemajuan teknologi dan industri baik di ranah nasional maupun internasional menyebabkan meningkatnya kebutuhan terhadap SDM ahli lebih khususnya programmer ahli. Akan tetapi dari sisi SDM dapat dikatakan sulit untuk menjadi seorang programmer ahli. Dalam penelitian yang dilakukan Wislow disebutkan bahwa butuh waktu 10 tahun bagi seorang programmer pemula untuk menjadi programmer ahli. Kelleher menyebutkan bahwa taksonomi tertinggi dalam lingkungan pemrograman/*programming environment* adalah *teaching system* (sistem pengajaran). Bagaimana dapat membentuk SDM yang ahli jika sistem pembelajaran tidak mendukung ?. Oleh karena itu peneliti mengambil studi kasus di kampus UIN Sunan Kalijaga Yogyakarta pada mata kuliah pemrograman terstruktur tahun 2012/2013. Model pembelajaran pada mata kuliah pemrograman terstruktur adalah menggunakan *C Free* (berbasis text). Peneliti mengusulkan model pembelajaran menggunakan IndoBlockly yaitu tool pemrograman visual untuk mempermudah pemahaman algoritma. IndoBlockly menggunakan puzzle sebagai media untuk memprogram dan tidak menggunakan text sehingga sangat meminimalisir *error syntax* yaitu masalah yang sering ditemui oleh programmer pemula.

Teknik pengumpulan data menggunakan pretes dan postes. Hasil dari analisis beda rata-rata untuk data pretes menghasilkan *p-value*  $0,749 > 0,005$  sehingga  $H_0$  diterima yaitu tidak ada perbedaan rata-rata pretes kelas eksperimen dan kelas kontrol itu menjadi bukti bahwa kelas kontrol dan kelas eksperimen keduanya homogen sebaliknya hasil dari analisis beda rata-rata postes kelas eksperimen dan kelas kontrol menghasilkan *p-value*  $0,000 < 0,005$  sehingga  $H_0$  ditolak dan disimpulkan bahwa rata-rata kelas eksperimen dan kelas kontrol berbeda. Hasil perhitungan gain juga diperoleh rata-rata gain kelas eksperimen sebesar 0,63 adalah gain sedang dan kelas kontrol 0,16 adalah gain rendah. Analisis korelasi menghasilkan koefisien korelasi sebesar 0,43 adalah kriteria sedang dengan *p-value*  $0,22 > 0,005$  yang berarti  $H_0$  diterima dan disimpulkan bahwa korelasi tidak signifikan. Hasil uji signifikansi koefisien korelasi memang menunjukkan bahwa korelasi tidak signifikan jadi korelasi hanya sampai taraf sedang. Hasil perhitungan determinasi yaitu  $r^2$  sebesar 0,1849 atau 18,49 % menunjukkan bahwa IndoBlockly hanya memberikan pengaruh 18,49 % terhadap peningkatan hasil belajar pemrograman terstruktur. Jadi model pembelajaran menggunakan IndoBlockly terbukti lebih baik dibandingkan dengan model pembelajaran konvensional walaupun hanya dapat memberikan pengaruh 18,49 % terhadap peningkatan hasil belajar.

Kata Kunci: *IndoBlockly, belajar pemrograman, pemahaman algoritma.*

## ABSTRACT

Rapid technological growth of computer technology and industry in the realm nationally and internationally cause increasing demands skilled of human resources expert in particular is expert programmers. However, we can be said to be an expert programmer is difficult . In a research Wislow mentioned that takes 10 years for a novice programmers to become expert programmers. Kelleher said that the highest taxonomic programming environment is the teaching system. How to become a expert if the system does not support for learning ?. Therefore i am a researcher took a studycase on campus UIN Sunan Kalijaga Yogyakarta on structured programming courses in 2012/2013. Model learning course on structured programming UIN Sunan Kalijaga used C Free (text-based). Researcher proposed a learning model that uses IndoBlockly visual programming tool that makes it easier to learn algorithms. IndoBlockly use the puzzle as a medium for programming and we not use text syntax so it minimizes the problems/ error syntax often encountered by novice programmers.

Data collection techniques using pretest and posttest method. The results of the analysis of average difference for data pretest  $p\text{-value } 0.749 > 0.005$  so  $H_0$  is accepted that there is no difference in the average pretest experimental class and control class it became evident that the experimental class and the control class both homogeneous otherwise result from the analysis of the mean difference The average posttest experimental class and control class produce  $p\text{-value } 0.000 < 0.005$  so  $H_0$  is rejected and it was concluded that the average experimental class and control class is different. The results of the calculation of the gain is also obtained an average gain of 0.63 experimental class is the gain medium and gain control class is a low 0.16. Correlation analysis resulting in the correlation coefficient of 0.43 was the criteria of being a  $p\text{-value } 0.22 > 0.005$  which means that  $H_0$  is accepted and concluded that the correlation is not significant. The results of significance test of correlation coefficient indicates that the correlation is not significant so only to some moderate correlation. The results of the calculation of determination  $r^2$  is 0.1849 or 18.49% indicate that IndoBlockly only give effect 18.49% increase in learning outcomes of structured programming. So using IndoBlockly proven learning model is better than conventional learning models although only 18.49% to give effect to the improvement of learning outcomes.

*Keywords: IndoBlockly, learn programming, algorithm comprehension.*

## BAB I

### PENDAHULUAN

#### 1.1 Latar belakang

Pembelajaran Praktikum Pemrograman Terstruktur di Jurusan Teknik Informatika UIN Sunan Kalijaga Yogyakarta saat ini masih menggunakan model konvensional.

Model pembelajaran saat ini adalah sebagai berikut:

1. Asisten menerangkan di depan kemudian mahasiswa memperhatikan.
2. Editor yang digunakan adalah *Turbo C* atau *C Free*.
3. Mahasiswa dituntut untuk menyalin *source code* C yang ada di modul.
4. Mahasiswa dituntut untuk meng-*compile source code* yang sudah disalin tadi dan jika ditemukan *error* maka mahasiswa akan bertanya kepada asisten.

Pada kenyataan di lapangan model seperti ini tidak berjalan dengan baik, kebanyakan mahasiswa hanya datang, duduk, mendengarkan penjelasan dari asisten kemudian menjadi tukang ketik yaitu menyalin *source code* yang ada di modul ke dalam editor C. Mahasiswa sering tidak paham apa yang mereka tulis, bagi mereka ketika setelah menulis *code* kemudian di-*compile* dan berjalan dengan *mulus* maka permasalahan selesai. Peneliti menilai bahwa model pembelajaran praktikum yang ada saat ini kurang efektif. Hal itu tidak hanya dirasakan oleh penulis tetapi juga para asisten Pemrograman Terstruktur dan juga dosen Pemrograman Terstruktur Teknik Informatika UIN Sunan Kalijaga.

Berbeda dengan model pembelajaran pemrograman di Negara maju. Penelitian yang dilakukan oleh Wislow menyebutkan bahwa butuh waktu 10 tahun bagi programmer pemula untuk menjadi programmer expert. Sistem pendidikan di Negara maju sudah mengantisipasi hal tersebut, bahasa pemrograman sudah mulai dikenalkan kepada anak-anak sejak TK(Taman Kanak-Kanak) dan SD(Sekolah Dasar) dalam bentuk permainan puzzle, permainan logika dan sebagainya, contohnya adalah Scratch , Greenfoot , App Inventor , dan Google Blockly. Software diatas menitikberatkan pada bagaimana caranya agar memprogram itu menyenangkan, software diatas dikemas dengan begitu menarik dan pengguna hanya perlu menyusun puzzle untuk membuat sebuah aplikasi atau program. Software diatas tidak hanya di peruntukkan untuk anak-anak TK dan SD, tapi diperuntukkan bagi siapa saja yang masih pemula dalam pemrograman. (Mafrur, 2012).

Pada studi kasus ini peneliti akan mencoba menerapkan model pembelajaran baru pada pembelajaran praktikum mata kuliah Pemrograman Terstruktur. Model pembelajaran yang baru ini kemudian akan dibandingkan dengan model konvensional yang saat ini masih berjalan dan nantinya akan ditarik kesimpulan apakah model pembelajaran yang baru ini bisa meningkatkan pemahaman mahasiswa terhadap mata kuliah Pemrograman Terstruktur atau tidak. Model pembelajaran yang peneliti usulkan hampir mirip dengan contoh model pembelajaran yang sudah peneliti sebutkan yaitu pembelajaran menggunakan *IndoBlockly*. *IndoBlockly* adalah sebuah aplikasi *open source* berbasis web yang dikembangkan oleh tim *IndoBlockly*. *IndoBlockly* sendiri

merupakan *Google Blockly* yang oleh tim *IndoBlockly* diterjemahkan menjadi berbahasa Indonesia dan ditambahkan berbagai fitur yang mendukung dengan pendidikan di Indonesia.

Rincinan model pembelajaran menggunakan *IndoBlockly* adalah sebagai berikut:

1. Editor yang digunakan adalah browser (Firefox, Chrome, Opera, Safari dll) editor *IndoBlockly* : <http://apps.developers.or.id/>
2. Mahasiswa mencoba menyelesaikan *maze (logic game)* yang ada di *IndoBlockly*
3. Mahasiswa langsung memulai membuat program dengan *IndoBlockly*
4. Mahasiswa tidak merasa seperti *coding* tetapi seperti bermain puzzle menggunakan *IndoBlockly*.
5. Program langsung bisa dijalankan dengan output dalam bentuk *dialog box javascript*
6. Blok-blok puzzle langsung bisa di-*generate* menjadi *source code C* yang langsung bisa di *compile* menggunakan *C compiler*.
7. Mahasiswa tidak disibukkan dengan *script* yang rumit (Mafrur, 2012).

Pada model pembelajaran menggunakan *IndoBlockly* ini yang peneliti tekankan adalah aspek pemahaman mahasiswa terhadap algoritma pemrograman karena tanpa pemahaman terhadap suatu algoritma mahasiswa tidak akan bisa membuat aplikasi dengan baik.

## 1.2 Rumusan Masalah

Apakah ada perbedaan terkait dengan pemahaman mahasiswa antara menggunakan pembelajaran model konvensional dan menggunakan *IndoBlockly*

pada mata kuliah Pemrograman Terstruktur di Teknik Informatika UIN Sunan Kalijaga Yogyakarta.

### **1.3 Batasan Masalah**

Batasan penelitian sbb:

1. Sampel yang diambil adalah mahasiswa kelas reguler semester I Teknik Informatika UIN Sunan Kalijaga tahun 2012/2013 dengan teknik *purposive sampling*.
2. Peneliti tidak melakukan proses karantina terhadap variabel kontrol maupun variabel percobaan sehingga bisa jadi ada berbagai faktor eksternal yang bisa mempengaruhi kedua variabel tersebut.
3. Peneliti tidak menghitung besaran factor external yang peneliti sebutkan pada point ke 3.
4. Proses pembelajaran dilakukan lima kali pertemuan penekanan pada pemahaman input output, variabel, array, kondisi, dan perulangan.

### **1.4 Tujuan Penelitian**

Untuk mengetahui apakah pembelajaran menggunakan *IndoBlockly* dapat meningkatkan pemahaman mahasiswa semester I tahun 2012/2013 terhadap mata kuliah Pemrograman Terstruktur dibandingkan menggunakan model pembelajaran konvensional.

### **1.5 Manfaat Penelitian**

Adapun manfaat yang dapat diambil dari penelitian ini adalah :

#### A. Bagi Pengembang IndoBlockly

Hasil dari penelitian ini akan dijadikan landasan dasar apakah IndoBlockly ini layak untuk dikembangkan atau tidak. Jika hasil dari penelitian ini menunjukkan bahwa IndoBlockly mempunyai pengaruh yang baik terhadap pemahaman mahasiswa untuk belajar pemrograman tentu ini akan dijadikan dasar bahwa IndoBlockly memang layak untuk dikembangkan lebih lanjut.

#### B. Bagi Mahasiswa/Umum

Hasil Penelitian ini akan membuktikan apakah memang IndoBlockly layak menjadi tool untuk belajar pemrograman baik bagi mahasiswa atau orang umum yang ingin belajar pemrograman.

#### C. Bagi Peneliti

Bagi peneliti, untuk menambah pengetahuan dan wawasan agar peneliti lebih terampil dalam penelitian khususnya yang melibatkan objek manusia secara langsung dan implementasi dari sebuah software. Hal itu disebabkan karena pada umumnya jurusan teknik informatika hanya berfokus pada penelitian pengembangan sistem tidak sepenuhnya implementasi ke end user.

#### D. Bagi Peneliti Selanjutnya

Karena disini peneliti juga termasuk pengembang dari IndoBlockly tentu dengan hasil penelitian ini jika memang hasilnya menunjukkan positif bahwa IndoBlockly memang bisa membantu siapapun yang ingin belajar pemrograman tentu peneliti akan lebih bersemangat untuk



mengembangkan IndoBlockly. Adapun jika hasilnya berkebalikan ataupun sama saja antara menggunakan IndoBlockly dan menggunakan model konvensional maka peneliti akan berusaha untuk mencari apa penyebabnya apakah karena IndoBlockly memang belum memenuhi kriteria sebagai software yang baik atau dari kesalahan implementasi.

### **1.6 Keaslian Penelitian**

Jenis penelitian seperti ini sudah banyak dilakukan khususnya oleh mahasiswa atau dosen dari jurusan pendidikan. Akan tetapi untuk penelitian implementasi dari sebuah tool kemudian menganalisis apakah tool tersebut dapat membantu pemahaman dalam mempelajari suatu hal dikarenakan menggunakan tool tersebut dalam pengetahuan kami penelitian seperti ini masih jarang. Disamping itu objek tool yang kami teliti adalah IndoBlockly yang peneliti juga ikut menjadi pengembang didalamnya jadi sudah pasti penelitian ini sama sekali belum pernah dilakukan sebelumnya.

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian-penelitian sebelumnya yang relevan dengan model penelitian ini cukup banyak. Model penelitian seperti ini yaitu penelitian pengujian suatu metode baru untuk membuktikan apakah metode tersebut lebih bagus dari metode yang lama sudah sering dilakukan terutama oleh kalangan pendidik. Yang berbeda dari penelitian ini adalah penelitian ini menguji software yang kami kembangkan oleh tim kami sendiri. Dari berbagai literatur yang penulis baca ada beberapa literatur yang relevan dengan penelitian kami.

Sebuah studi literatur yang berkaitan dengan psikologi dan pendidikan pemrograman yang dilihat dari persepektif seorang pengajar telah dilakukan oleh Robin dan kawan-kawannya. Tujuan penelitiannya adalah untuk mengidentifikasi permasalahan-permasalahan dalam proses belajar programmer pemula. Ada beberapa hal yang membuat kami tertarik seperti yang tercantum dalam studi literturnya yaitu penelitian dari (Guindon ,1990) yang menyebutkan karakter dari seorang ahli adalah memiliki skema pengetahuan efisien terorganisir dan khusus, paham dalam algoritma yang mendasar (bukan rincian dangkal seperti sintaks bahasa). Kesimpulan dari penelitian Robins dan kawan-kawannya adalah sifat yang mendasari apa yang membuat seorang pemula yang efektif? Bagaimana kita dapat mengubah efektif pemula menjadi lebih efektif?. Berbagai faktor yang berpotensi relevan yaitu motivasi, kepercayaan diri atau tanggapan emosional,

aspek umum atau pengetahuan khusus, dan strategi. Jadi disamping faktor dari dalam diri programmer itu sendiri juga ada faktor dari luar yaitu strategi (strategi pengajaran) yang bisa mempercepat proses menjadikan programmer pemula menjadi programmer ahli (Robins dkk, 2003).

Selanjutnya adalah Penelitian dari Lahtinen dan kawan-kawannya. Penelitian ini merupakan survei internasional dengan responden 500 siswa dan guru. Survei ini memberikan informasi tentang kesulitan yang dirasakan ketika belajar dan mengajar pemrograman. Hasil survei juga dapat dijadikan sebagai dasar untuk rekomendasi dan *developing* bahan pembelajaran. Dalam penelitian tersebut disimpulkan bahawa kesulitan yang biasa di alami baik siswa maupun guru dalam belajar dan mengajar pemrograman tidak hanya konsep pemrograman yang abstrak yang cukup susah untuk diinterpretasikan dalam dunia nyata akan tetapi konstruksi program juga termasuk didalamnya. Penelitian tersebut dilakukan menggunakan dua bahasa pemrograman sample yaitu C++ dan Java. Jadi selain konsep pemrograman yang abstrak, susunan bahasa / konstruksi dari bahasa yang di pelajari itu juga mempengaruhi apakah bahasa itu sulit atau mudah untuk dipelajari (Lahtinen dkk, 2005).

Yang ketiga yaitu Penelitian yang dilakukan oleh Kelleher dan Pausch. Dalam penelitiannya Kelleher dan Pausch menyajikan taksonomi bahasa dan lingkungan yang dirancang untuk membuat pemrograman lebih mudah diakses untuk programmer pemula dari segala usia. Penelitian ini menjelaskan semua kategori dalam taksonomi tersebut, memberikan gambaran singkat tentang sistem dalam setiap kategori, dan menyarankan beberapa jalan keluar untuk masa depan

dalam lingkungan pemrograman pemula dan bahasa pemrograman .Penelitian tersebut membahas satu pertanyaan pokok yaitu dalam menciptakan lingkungan pemrograman untuk pemula, salah satu pertanyaan pertama yang harus dijawab adalah mengapa siswa/mahasiswa perlu memprogram?. Ada berbagai motivasi yang mendasari mengapa siswa/mahasiswa belajar memprogram yaitu : pemrograman sebagai karir, memprogram untuk belajar bagaimana memecahkan masalah dengan cara yang terstruktur dan logis, untuk membangun perangkat lunak untuk keperluan pribadi, untuk mengeksplorasi ide-ide dalam bidang studi lainnya, dan lain sebagainya.

Pada Tulisan Kelleher dan Pausch ini taksonomi atau urutan tertinggi pada lingkungan pemrograman/*programming environment* adalah *teaching system* (sistem pengajaran) dan *Empowering Systems* (pemberdayaan sistem) kemudian disusul dengan berbagai hirarki taksonomi dibawahnya. *Teaching system* dan *Empowering Systems* sebagai hirarki yang paling tinggi, jelas bagaimana seorang dapat memahami pemrograman jika dia tidak belajar, dan bagaimana dia belajar jika tidak diajar. Kemudian disusul hirarki seperti *Mechanics of programming* (sisi teknis dari pemrograman) dan seterusnya. Dalam tulisannya juga disebutkan berbagai bahasa pemrograman yang bisa mempermudah siapapun yang ingin belajar pemrograman dan bisa digunakan dari segala usia. Disamping itu dia juga menyebutkan berbagai kendala dan hambatan bagi siapa saja yang belajar bahasa pemrograman. Kendala tersebut ada dua yaitu kendala teknis dan kendala sosial.

Berbagai kendala teknis seperti konsep yang abstrak, syntaks yang susah untuk dihafal dan sebagainya, untuk kendala teknis ini sudah banyak dibahas oleh

banyak peneliti. Beberapa peneliti mengajukan konsep dan design seperti pemrograman berbasis visual, game, dan sebagainya. Begitu juga dengan penelitian Kelleher dan Pausch ini mereka memberikan banyak contoh-contoh bahasa pemrograman visual yang dapat mempermudah pemula dalam memahami bahasa pemrograman. Kendala yang kedua yaitu kendala sosial. Analisis Kelleher dan Pausch menyebutkan bahwa ternyata tidak hanya kendala teknis yang dihadapi oleh programmer pemula yang ingin belajar pemrograman tapi juga ada kendala sosial. Contoh kendala sosial adalah siswa/mahasiswa memutuskan dan memilih untuk tidak belajar pemrograman. Kelleher dan Pausch mengatakan bahwa kendala sosial ini lebih sulit untuk diselesaikan daripada kendala teknis. Solusi dari kendala sosial ini adalah sosialisai dan dukungan sosial. Kelleher dan Pausch menyimpulkan beberapa solusi yang bisa digunakan untuk mengatasi kendala-kendala baik itu teknis maupun sosial dalam belajar pemrograman yaitu: menyederhanakan mekanisme pemrograman, memberikan dukungan bagi pelajar, dan memotivasi siswa untuk belajar memprogram. Dalam hal mengatasi hambatan sosial bisa dengan mendukung peserta didik atau memberikan alasan yang menarik untuk belajar pemrograman (Kelleher dan Pausch, 2003 ).

Dari penelitian Kelleher dan Pausch ini respon dari penulis adalah bahwa kendala-kendala itu memang benar-benar nyata. Hal tersebut bisa penulis utarakan berdasarkan pengalaman penulis sendiri baik ketika penulis belajar pemrograman maupun pengalaman penulis ketika mengajar pemrograman. Kendala teknis adalah kendala yang akan pertama dihadapi oleh siapapun yang ingin belajar pemrograman, oleh karena itu penulis mengusulkan sebuah konsep dan design

yang sebenarnya sudah banyak diusulkan dan diimplementasikan sejak dahulu akan tetapi belum di implementasikan dalam lingkungan penulis yaitu VPLs(*Visual Programming Languages*).

Penelitian yang relevan ke empat yaitu paper dari scratch yang dilakukan oleh Maloney dan kawan-kawannya. Maloney dan kawan-kawannya ini menjelaskan scratch yaitu bahasa pemrograman visual blok yang dirancang dan difasilitasi berbagai media untuk programmer pemula. Pada penelitian ini Maloney dan kawan-kawannya membentuk sebuah Clubhouse yang di peruntukkan bagi siapa saja untuk datang ke Clubhouse. Penelitian tersebut dilakukan selama periode 18 bulan, setelah periode 18 bulan kemudian Maloney menganalisis 34 % atau sebanyak 536 proyek-proyek scratch yang ada di Clubhouse tadi. Dalam paper tersebut juga dibahas mengenai motivasi pemuda perkotaan yang memilih untuk memprogram menggunakan scratch daripada menggunakan salah satu dari banyak paket perangkat lunak lain yang tersedia bagi mereka. Ada yang menurut penulis cukup menarik disini. Pada tulisan Maloney dkk di jelaskan sebagai berikut: Pertanyaan yang lebih mendesak adalah mengapa pemuda memilih untuk terlibat dalam memprogram menggunakan scratch di Scratch Clubhouse mengingat bahwa mereka memiliki banyak pilihan perangkat lunak lain? Jawaban terbaik mungkin sudah dijawab oleh Kelleher dan Pausch yang mencatat bahawa sistem dapat membuat pemrograman menjadi lebih mudah diakses oleh pemula dan dikarenakan scratch dapat menyederhanakan mekanisme pemrograman, dengan memberikan dukungan bagi pelajar, dan dengan memberikan siswa dengan motivasi untuk belajar memprogram.

Maloney menyatakan kami berpikir bahwa scratch mempunyai tiga hal yang menarik. Yang pertama, desain blok scratch menyederhanakan mekanisme pemrograman dengan meminimalisir kesalahan sintaks, scratch langsung dapat memberikan umpan balik mengenai penempatan blok, dan scratch langsung dapat memberikan umpan balik untuk percobaan/ketika program dijalankan. Selain itu, kita berpikir bahwa infrastruktur sosial Komputer Clubhouse penting dalam memberikan dukungan bagi programmer pemula. Jawaban tersebut jelas masuk akal karena di scratch Clubhouse kami juga menyediakan beberapa mentor yang bisa mengajarkan para programmer pemula, dan juga mungkin dengan seringnya mereka berada di scratch Clubhouse mereka bisa bertukar program dan ide dengan teman-teman mereka (Maloney dkk, 2008).

Dari beberapa penelitian yang relevan dengan penelitian ini yang sudah penulis sampaikan diatas yang paling mendekati dengan penelitian yang penulis lakukan ini adalah penelitian ke empat yang dilakukan oleh Maloney dan kawan-kawannya. Tentu banyak yang mengetahui tentang scratch dan begitu mudahnya memprogram dengan menggunakan scratch. Penelitian yang penulis lakukan jelas masih jauh dari penelitian ini, kalau penelitian yang dilakukan Maloney dan kawan-kawannya memang akan memberikan hasil yang general karena sampel yang digunakan berupa proyek-proyek general dari berbagai anak/siswa yang sering *nongkrong*/bermain di scratch ClubHouse. Proyek scratch memang tidak main-main scratch berani membuat sebuah ClubHouse demi untuk kelangsungan dan analisis penelitian proyek scratch itu sendiri dan disini inilah yang belum bisa penulis lakukan. Semoga pada penelitian IndoBlockly berikutnya akan lebih baik.

## **2.2 Landasan Teori**

Landasan teori menjelaskan berbagai teori yang berhubungan dengan penelitian yang dilakukan oleh penulis baik yang berhubungan dengan pendidikan dan pembelajaran, kemudian pemrograman dan juga software yang digunakan sebagai objek penelitian.

### **2.2.1 Pengertian Belajar**

Belajar adalah suatu aktivitas mental/psikis yang berlangsung dalam interaksi aktif dengan lingkungan yang menghasilkan perubahan dalam pengetahuan, pemahaman, keterampilan dan nilai sikap (Winkel, 1983). Sedangkan menurut (Gagne dalam Dahar, 1989) belajar adalah suatu proses di mana suatu organisme berubah perilakunya sebagai akibat dari pengalaman. Dengan belajar tindakan perilaku siswa akan berubah ke arah yang lebih baik. Berhasil baik atau tidaknya belajar tergantung dari faktor-faktor yang mempengaruhinya. Faktor-faktor tersebut terdiri dari faktor internal, eksternal dan pendekatan belajar.

1. Faktor internal adalah faktor dari dalam diri siswa, yaitu keadaan/kondisi jasmani dan rohani siswa meliputi aspek fisiologis (kondisi tubuh dan panca indera), dan aspek psikologis antara lain: intelegensi dalam, sikap misalnya dalam beradaptasi dengan teman, bakat dalam mengerjakan soal, minat dalam mengikuti pelajaran serta punya kemauan besar untuk belajar dan mempunyai motivasi untuk belajar baik individu maupun dalam kelompok.



2. Faktor eksternal adalah faktor dari luar diri siswa, yaitu kondisi lingkungan di sekitar siswa meliputi faktor lingkungan sosial (guru, teman, masyarakat, dan keluarga) dan faktor lingkungan non-sosial (gedung, sekolah, tempat tinggal, alat belajar, cuaca dan waktu belajar) .

### **2.2.2 Media Pembelajaran**

Kata media berasal dari bahasa latin *Medius* yang secara harfiah berarti '*perantara*' atau '*pengantar*'. Menurut Boove (dalam Ena, 2007), media adalah sebuah alat yang berfungsi untuk menyampaikan pesan pembelajaran. Pembelajaran adalah sebuah komunikasi antara pembelajar, pengajar dan bahan ajar. Komunikasi tidak akan berjalan tanpa bantuan sarana penyampai pesan atau media. Bentuk-bentuk stimulus yang bisa dipergunakan sebagai media diantaranya adalah hubungan atau interaksi manusia, realita, gambar bergerak atau tidak, tulisan dan suara yang direkam (Ena, 2007).

Menurut Gerlach dan Ely (dalam Arsyad, 2002), ciri-ciri media ada tiga, yaitu.

1. Ciri Fiksatif (Fixative Property)

Ciri ini menggambarkan kemampuan media merekam, menyimpan, melestarikan, dan merekonstruksi suatu peristiwa atau objek.

2. Ciri Manipulatif (Manipulatif Property)

Transformasi suatu kejadian atau objek dimungkinkan karena media memiliki ciri manipulatif. Kejadian yang memakan waktu berhari-hari dapat disajikan kepada siswa dalam waktu dua atau tiga menit dengan teknik pengambilan gambar *time-lapse recording*.

### 3. Ciri Distributif (Distributive Property)

Ciri distributif dari media memungkinkan suatu objek atau kejadian ditransportasikan melalui ruang, dan secara bersamaan kejadian tersebut disajikan kepada sejumlah besar siswa dengan stimulus pengalaman yang relatif sama mengenai kejadian itu.

Fungsi utama media pembelajaran adalah sebagai alat bantu mengajar yang ikut mempengaruhi iklim, kondisi, dan lingkungan belajar yang ditata dan diciptakan oleh guru. Sedangkan manfaat penggunaan media (Arsyad, 2002), antara lain:

1. Media pembelajaran dapat memperjelas penyajian pesan dan informasi sehingga dapat memperlancar dan meningkatkan proses dan hasil belajar.
2. Media pembelajaran dapat meningkatkan dan mengarahkan perhatian anak sehingga dapat menimbulkan motivasi belajar, interaksi yang lebih langsung antara siswa dan lingkungannya, dan kemungkinan siswa untuk belajar sendiri-sendiri sesuai dengan kemampuan dan minatnya.
3. Media pembelajaran dapat mengatasi keterbatasan indera, ruang, dan waktu.
4. Media pembelajaran dapat memberikan kesamaan pengalaman kepada siswa tentang peristiwa-peristiwa di lingkungan mereka, serta memungkinkan terjadinya interaksi langsung.

### **2.2.3 Proses Pembelajaran**

Penilaian adalah upaya atau tindakan untuk mengetahui sejauh mana tujuan yang telah ditetapkan itu tercapai atau tidak. Penilaian berfungsi sebagai

alat untuk mengetahui keberhasilan proses dan hasil belajar siswa/mahasiswa. Dalam sistem pendidikan nasional rumusan tujuan pendidikan, baik tujuan kurikuler maupun tujuan instruksional, menggunakan klasifikasi hasil belajar dari Benyamin Bloom yang secara garis besar membaginya menjadi tiga ranah, yakni ranah kognitif, ranah afektif, dan ranah psikomotorik (Bloom, 1956).

Salah satu prinsip dasar yang harus senantiasa diperhatikan dalam pendidikan adalah melaksanakan evaluasi hasil belajar secara menyeluruh terhadap peserta didik, baik dari segi pemahamannya terhadap materi atau bahan ajar yang telah diberikan (aspek kognitif), maupun dari segi penghayatan (aspek afektif), dan pengamalannya (aspek psikomotor). Benjamin S. Bloom dan kawan-kawannya itu berpendapat bahwa pengelompokkan tujuan pendidikan itu harus senantiasa mengacu kepada tiga jenis domain (ranah) yang melekat pada diri peserta didik, yaitu:

- a) Ranah proses berfikir (*cognitive domain*)
- b) Ranah nilai atau sikap (*affective domain*)
- c) Ranah keterampilan (*psychomotor domain*)

Dalam konteks evaluasi hasil belajar, maka ketiga domain atau ranah itulah yang harus dijadikan sasaran dalam setiap kegiatan evaluasi hasil belajar. Sasaran kegiatan evaluasi hasil belajar adalah:

1. Apakah peserta didik sudah dapat memahami semua bahan atau materi pelajaran yang telah diberikan pada mereka?

2. Apakah peserta didik sudah dapat menghayatinya?
3. Apakah materi pelajaran yang telah diberikan itu sudah dapat diamalkan secara kongkret dalam praktek atau dalam kehidupannya sehari-hari?

Diantara ketiga ranah tersebut, ranah kognitiflah yang paling banyak dinilai oleh para dosen /guru karena berkaitan dengan kemampuan para siswa dalam menguasai isi bahan pengajaran.

#### **2.2.3.1 Pengertian Ranah Penilaian Kognitif**

Ranah kognitif adalah ranah yang mencakup kegiatan mental (otak). Menurut Bloom, segala upaya yang menyangkut aktivitas otak adalah termasuk dalam ranah kognitif. Ranah kognitif berhubungan dengan kemampuan berfikir, termasuk didalamnya kemampuan menghafal, memahami, mengaplikasi, menganalisis, mensintesis, dan kemampuan mengevaluasi. Dalam ranah kognitif itu terdapat enam aspek atau jenjang proses berfikir, mulai dari jenjang terendah sampai dengan jenjang yang paling tinggi. Keenam jenjang atau aspek yang dimaksud adalah:

- a. Pengetahuan/hafalan/ingatan (*knowledge*)

Adalah kemampuan seseorang untuk mengingat-ingat kembali (recall) atau mengenali kembali tentang nama, istilah, ide, rumus-rumus, dan sebagainya, tanpa mengharapkan kemampuan untuk menggungkannya. Pengetahuan atau ingatan adalah merupakan proses berfikir yang paling rendah. Salah satu contoh hasil belajar kognitif pada jenjang pengetahuan adalah dapat menghafal jenis-jenis variabel yang biasa digunakan dalam memprogram dsb.

b. Pemahaman (*comprehension*)

Adalah kemampuan seseorang untuk mengerti atau memahami sesuatu setelah sesuatu itu diketahui dan diingat. Dengan kata lain, memahami adalah mengetahui tentang sesuatu dan dapat melihatnya dari berbagai segi. Seseorang peserta didik dikatakan memahami sesuatu apabila ia dapat memberikan penjelasan atau memberi uraian yang lebih rinci tentang hal itu dengan menggunakan kata-katanya sendiri. Pemahaman merupakan jenjang kemampuan berfikir yang setingkat lebih tinggi dari ingatan atau hafalan. Salah satu contoh hasil belajar ranah kognitif pada jenjang pemahaman ini adalah mahasiswa bisa paham terhadap algoritma searching kemudian dapat menguraikannya kembali dengan bahasa mereka sendiri.

c. Penerapan (*application*)

Adalah kesanggupan seseorang untuk menerapkan atau menggunakan ide-ide umum, tata cara ataupun metode-metode, prinsip-prinsip, rumus-rumus, teori-teori dan sebagainya, dalam situasi yang baru dan kongkret. Penerapan ini adalah merupakan proses berfikir setingkat lebih tinggi ketimbang pemahaman. Salah satu contoh hasil belajar kognitif jenjang penerapan adalah mahasiswa mampu menerapkan algoritma yang sudah ia pahami ke dalam bentuk code dengan bahasa pemrograman yang ia kuasai.

d. Analisis (*analysis*)

Adalah kemampuan seseorang untuk merinci atau menguraikan suatu bahan atau keadaan menurut bagian-bagian yang lebih kecil dan mampu

memahami hubungan di antara bagian-bagian atau faktor-faktor yang satu dengan faktor-faktor lainnya. Jenjang analisis adalah setingkat lebih tinggi ketimbang jenjang aplikasi. Contoh dari hasil belajar kognitif jenjang analisis adalah mahasiswa mampu menganalisis algoritma dan *source code*, misalnya terjadi error ketika di jalankan maka mahasiswa dalam tahap analisis ini harus mampu menyelesaikan permasalahan tersebut (*debugging*) sampai program bisa benar-benar berjalan dengan baik.

e. Sintesis (*syntesis*)

Adalah kemampuan berfikir yang merupakan kebalikan dari proses berfikir analisis. Sintesis merupakan suatu proses yang memadukan bagian-bagian atau unsur-unsur secara logis, sehingga menjelma menjadi suatu pola yang berstruktur atau bebrbentuk pola baru. Jenjang sintesis kedudukannya setingkat lebih tinggi daripada jenjang analisis. Contoh: mahasiswa tidak hanya mampu memahami algoritma kemudian menuliskan dalam *source code* dan juga melakukan *debugging* tetapi mahasiswa juga mampu untuk memodifikasi atau membuat algoritma baru.

f. Penilaian (*evaluation*)

Penilaian merupakan jenjang berpikir paling tinggi dalam ranah kognitif dalam taksonomi Bloom. Penilaian/evaluasi disini merupakan kemampuan seseorang untuk membuat pertimbangan terhadap suatu kondisi, nilai atau ide, misalkan jika seseorang dihadapkan pada beberapa pilihan maka ia akan mampu memilih satu pilihan yang terbaik sesuai dengan patokan-patokan atau kriteria

yang ada. Salah satu contoh hasil belajar kognitif jenjang evaluasi adalah: mahasiswa mampu untuk menilai dan memilih algoritma yang paling efektif yang akan dia gunakan. Keenam jenjang berpikir yang terdapat pada ranah kognitif menurut Taksonomi Bloom itu, jika diurutkan secara hirarki piramidal adalah sebagai tertulis pada Gambar 2.1.



**Gambar 2.1 Hirarki Piramida Taksonomi Bloom**

#### **2.2.3.2 Ranah Afektif dan Psikomotorik**

Pemrograman komputer memang lebih banyak berkaitan dengan ranah kognitif akan tetapi bukan berarti kedua ranah atau domain yang lain itu dihilangkan. Ranah afektif adalah ranah penghayatan kemudian disusul dengan ranah psikomotorik atau ranah skill. Proses belajar pemrograman itu semua berada pada ranah kognitif dimulai dari menghafal sintaks, memahami, menganalisis kesalahan, mensintesis algoritma baru, menilai dan memilih algoritma yang paling efektif dan sebagainya.

Ranah afektif akan ada ketika mahasiswa yang tadinya dia tidak mengetahui apa-apa mengenai pemrograman, kemudian dia sudah melalui ranah kognitif dan akhirnya jiwa programmer secara tidak sadar akan terbentuk di dalam diri mahasiswa tersebut. Ketika penghayatan sebuah algoritma dan juga penghayatan dalam menulis kode dalam proses selanjutnya ini akan menuju ranah psikomotorik yaitu ranah skill. Ranah skill adalah ranah dimana yang tadinya hanya seorang mahasiswa atau programmer pemula akan mempunyai skill yang cukup hebat yang menjadikan dirinya menjadi programmer ahli.

#### **2.2.4 Belajar Pemrograman (*Learning Programming*)**

Pemrograman adalah keterampilan yang sangat berguna dan dapat menjadi karir. Baru-baru ini kebutuhan akan permintaan terhadap programmer dan minat siswa dalam pemrograman berkembang pesat, dan kursus pemrograman telah menjadi semakin populer. Belajar pemrograman membutuhkan usaha yang keras bagi seorang programmer pemula biasanya mengalami berbagai kesulitan. Kursus/Sekolah Pemrograman umumnya dianggap sulit, dan sering memiliki risiko tingkat putus sekolah paling tinggi (Winslow, 1996). Dalam penelitian disebutkan bahwa dibutuhkan sekitar 10 tahun untuk mengubah programmer pemula menjadi programmer ahli (Robins dkk, 2003).

Pemahaman dalam memprogram merupakan bagian penting dari keterampilan pemrograman komputer, baik dari yang praktis, perspektif maupun teoritis. Ini adalah sebuah keterampilan kognitif yang kompleks, yang melibatkan akuisisi representasi mental dari struktur program dan fungsi. Dari sudut pandang



Theoretical , pemahaman melibatkan penugasan makna tertentu yaitu sesuatu yang membutuhkan pengetahuan khusus. Dari sudut pandang praktis, kemampuan untuk memahami program yang ditulis oleh orang lain atau ketika membuat program sendiri adalah komponen penting keahlian seorang programmer. Skill/keahlian pemahaman ini akan digunakan programmer untuk melakukan tugas-tugas pemrograman seperti *debugging*, *editing* dan, *code reuse*. (Navarro-Prieto, 2000).

Pennington Memperkirakan bahwa lebih dari 50% dari semua waktu programmer profesional dihabiskan untuk tugas-tugas pemeliharaan program yang melibatkan modifikasi/editing dan update dari program yang sebelumnya ditulis. Dari berbagai hal diatas dapat kita simpulkan bahwa pemahaman memainkan peran sentral dalam pemrograman. Oleh karena itu, penelitian tentang strategi pemahaman akan sangat berguna karena hasilnya dapat memberikan informasi yang berguna untuk meningkatkan kinerja programmer, kemajuan pendidikan, kemajuan teknologi desain dan lingkungan pemrograman (*programming environments*) (Pennington, 1987a).

Dari berbagai referensi diatas kita dapat menyimpulkan bahawa belajar pemrograman itu memang susah terutama bagi programmer pemula/ mahasiswa yang sedang belajar pemrograman. Sperti penelitian yang dilakukan oleh Wislow bahwa bagai programmer pemula butuh 10 tahun untuk menjadi programmer yang ahli. Untuk menjadi programmer yang baik, mahasiswa membutuhkan skill dan usaha yang keras untuk menghafalkan sintak dan memahami aturan belajar pemrograman itu sendiri (Esteves dan Mendes, 2004). Mahasiswa harus tahu dan

paham bagaimana menyelesaikan suatu masalah dan belajar membuat dan merinci algoritma yang efisien itulah yang menyebabkan belajar pemrograman terkesan sulit. Pada Kenyataanya banyak mahasiswa yang gagal dan menemui masalah ini pada awal pelajaran/mata kuliah pemrograman untuk pertama kalinya. Sudah banyak peneliti yang meneliti masalah ini (Lahtinen dkk, 2005). Salah satu penyebab yang disebutkan adalah mahasiswa tidak siap untuk berfikir dan menyelesaikan masalah. Pada proses belajar mengajar sebelumnya mahasiswa tidak terbiasa untuk menyelesaikan suatu masalah/kasus, sehingga mahasiswa tidak mempunyai kemampuan problem solving yang baik (Gomes, 2006). Beberapa mahasiswa yang cukup mampu menyelesaikan masalah dan membuat algoritma biasanya juga terkendala dalam masalah waktu penyelesaian.

Penyebab lain dari kegagalan belajar pemrograman adalah mahasiswa sudah mengatakan bahwa dia tidak menyukai pemrograman dikarenakan konsepnya susah untuk dipahami, seperti variabel, tipe data, alamat memori dikarenakan itu adalah konsep yang abstrak yang tidak bisa direpresentasikan di dunia nyata (Miliszewska, 2007). Penyebab lain adalah proses pembelajaran yang cenderung masih tradisional yaitu dosen hanya mengajarkan sintak bahasa pemrograman tertentu dan tidak matang dalam hal pemahaman konsep (Lethbridge, 2007). Dengan kemajuanya teknologi dan perkembangan software yang begitu pesat beberapa peneliti mengusulkan design dan software pemrograman berbasis visual dan memang kenyataanya sekarang banyak software-software yang diciptakan dengan tujuan untuk mengatasi masalah-masalah diatas. Memang untuk saat ini masih dominan software pemrograman

yang berupa text based, tetapi ada juga yang visual based bahkan ada yang berbentuk game (Hundhausen, 2007).

#### **2.2.5 Bahasa Pemrograman Visual (*Visual Programming Language /VPLs*)**

Dari mulai lukisan di gua hieroglif sejak dulu manusia sudah lama berkomunikasi satu dengan lainnya menggunakan gambar. Banyak para peneliti di bidang VPL bertanya: Mengapa kemudian kita saat ini berkomunikasi dengan komputer menggunakan pemrograman tekstual? Bukankah akan lebih produktif jika dapat menginstruksikan komputer hanya menggunakan gambar.? Jelas para pendukung VPLs menjawab ya. Pertanyaan-pertanyaan diatas menjadi motivasi utama para peneliti di bidang VPLs. Pertama. banyak orang berpikir dan mengingat hal hal dari segi gambar. Selain itu pemrograman tekstual telah terbukti menjadi masalah bagi orang-orang kreatif dan cerdas , mereka harus banyak belajar terlebih dahulu mengenai sintak-sintak tekstualnya dari bahasa pemrograman itu sendiri. Permasalahan yang lain adalah ide lebih mudah diterjemahkan dalam bentuk gambar visual atau grafik akan tetapi kita harus menerjemahkan ide tersebut kedalam bentuk tekstual.Selain itu banyak bidang-bidang ilmu yang memerlukan simulasi dalam bentuk visual dan tidak bisa di interpretasikan dalam bentuk tekstual. Dari berbagai alasan diatas itu membuktikan bahawa penelitian mengenai VPLs itu menarik dan penting (Boshernitsan dan Downes, 2004)

### **2.2.5.1 Kelebihan dan Kekurangan Bahasa Pemrograman Visual**

Pemrograman grafis/visual memiliki kelebihan dan kekurangan dibandingkan dengan pemrograman tekstual. Landasan teori mengenai bahasa pemrograman visual ini kami mengambil dari tulisan.

#### **2.2.5.1.1 Kelebihan VPLs**

Andrew Begel mengatakan bahwa dengan menggunakan representasi grafis dari benda-benda anda bisa lebih konkret merepresentasikan suatu objek (klik dua kali pada objek koper dan melihat apa yang ada di dalam), menghilangkan sintak-sintak yang susah untuk dipahami (seperti {} dan () dalam C, BEGIN dan END dan () dalam Pascal, dan () dalam Lisp) (Begel, 1996).

Pemrograman grafis/visual juga dapat dikatakan sebagai metafora dari kehidupan nyata untuk membuat pemrograman lebih mudah. Sebagai contoh, pemrograman tombol lampu untuk menghidupkan dan mematikan lampu. Pemrograman grafis juga memungkinkan lebih mudah untuk berbagi program. Anda dapat mendefinisikan program Anda menjadi blok tertentu dan hanya "memberikan" blok ke teman untuk mencoba sendiri. Hal ini mirip dengan manfaat yang diusulkan desain OOP dalam bahasa tekstual. Keuntungan lain adalah mudah dipahami. Melihat gambar dari sebuah program, pengguna dapat lebih mudah membedakan maknanya, daripada melihat program tekstual yang terdiri dari file dan kode yang sangat banyak. Mungkin salah satu keuntungan terbaik adalah penggunaan isyarat visual dalam bahasa

grafis/visual. Sambungan/koneksi antar objek dapat dibuat lebih eksplisit melalui desain dan grafis daripada menggunakan tekstual.

#### **2.2.5.2 Kekurangan VPLs**

Disamping ada kelebihan di sisi lain juga ada kerugian dari VPLs. Beberapa bahasa grafis/visual yang dapat menyebabkan frustrasi bagi programmer expert yang ingin mengungkapkan pernyataan yang mungkin lebih baik dan efektif diwakili menggunakan teks. Selain itu Masalah pemrograman visual adalah bahwa Anda tidak bisa memiliki lebih dari 50 primitif visual pada layar pada saat yang sama. "Pertanyaan selanjutnya adalah Bagaimana Anda akan menulis sebuah sistem operasi dengan pemrograman visual?" Masalah lain adalah extendibility bahasa. C dan Lisp dibangun untuk dikembangkan oleh para programmer. Pada saat ini, bahasa grafis cenderung terbatas pada pembuat desain tanpa berpikir untuk menambahkan fitur tambahan.

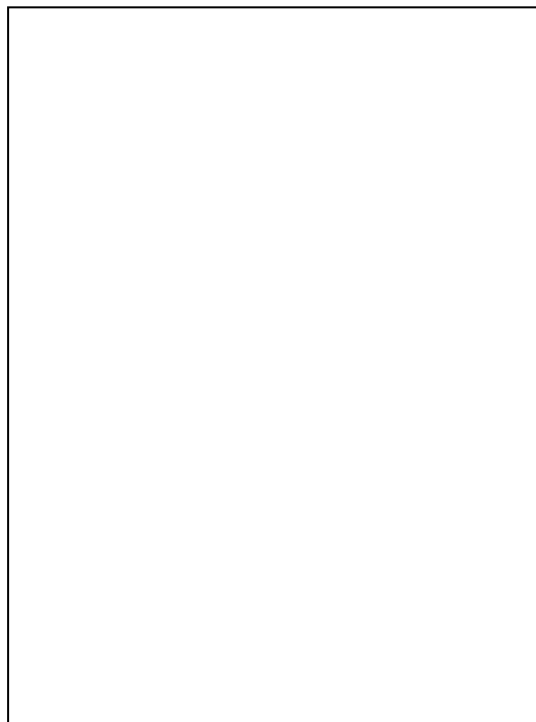
Ada daya tarik yang sangat tinggi dari Pemrograman Visual yaitu untuk anak-anak. Anak-anak suka memanipulasi blok dan mengumpulkan koleksi benda. seringkali masalah utama untuk mengajarkan pemrograman kepada anak-anak adalah bahwa sintak-sintak pemrograman yang sulit dipahami. VPLs menghilangkan masalah ini, anak-anak akan lebih mudah untuk membuat program. Akan tetapi Sebuah bahasa grafis/visual yang terlalu sederhana dapat mengurangi kreatifitas seorang anak untuk membuat program/sesuatu yang menyenangkan. Ketika anak bertambah usia dan lebih cerdas, ia dapat bermigrasi

ke bahasa tekstual dan menggunakan kompleksitas yang lebih besar (Begel, 1996).

### **2.2.6 IndoBlockly**

IndoBlockly mirip dengan bahasa visual block sebelumnya seperti *scratch*, *Greenfoot*, *AppInventor*, dan *Google Blockly*. Dengan menggunakan IndoBlockly pengguna tidak akan merasa coding tetapi yang dirasakan adalah menyusun puzzle. “*Semua bisa jadi programer dengan IndoBlockly*”, itu adalah slogan IndoBlockly, IndoBlockly dilengkapi dengan generator bahasa C dan Pascal, blok-blok yang sudah disusun oleh pengguna langsung dapat diubah menjadi source code C atau Pascal (Mafrur, 2012).

#### **2.2.6.1 Konsep IndoBlockly**



**Gambar 2.2 . Konsep IndoBlockly**

Kriteria software sesuai dengan konsep IndoBlockly adalah :

1. Berbahasa Indonesia.
2. Menarik, tidak membosankan, user tidak terasa seperti coding tetapi seperti bermain.
3. Meminimalisir penggunaan sintak-sintak yang susah untuk dimengerti oleh pengguna baru.
4. IDE yang portabel (cloud/web based).
5. IDE yang mempunyai tampilan seperti IDE Desktop Environment.
6. Hasilnya bisa dikonvert ke dalam source code C atau Pascal dan langsung bisa di eksekusi dengan C atau Pascal compailer
7. Format file penyimpanan yang portabel.
8. Ada beberapa game logika untuk mengasah kemampuan otak anak.
9. Bisa meningkatkan pemahaman user terhadap konsep pemrograman dan algoritma.

Pada awal tahun 2012 Google meluncurkan *Google Blockly* dengan lisensi opensource. Kemudian kami bersepakat untuk mengambil source dari *Google Blockly* kemudian kami modifikasi sesuai dengan konsep awal tadi. Nama IndoBlockly juga diambil dari *Google Blockly*, Indo adalah Indonesia dan Blokly adalah blok-blok puzzle dari *Google Blockly*.

#### **2.2.6.2 Design Interface IndoBlockly**

Saat ini IndoBlockly belum sepenuhnya mendukung semua fitur-fitur yang ada dalam tool bahasa pemrograman. Oleh sebab itu pada batasan masalah

peneliti juga membatasi bahwa yang penulis teliti adalah hasil belajar pemrograman terstruktur meliputi pemahaman terhadap input output, variabel, array, kondisi dan perulangan. Pada pembahasan ini kami menyertakan sreenshoot dari aplikasi indoBlockly.

## 1. Index IndoBlockly

Gambar 2.2 adalah gambar tampilan awal tool IndoBlockly yang bisa di akses di <http://apps.developers.or.id> .



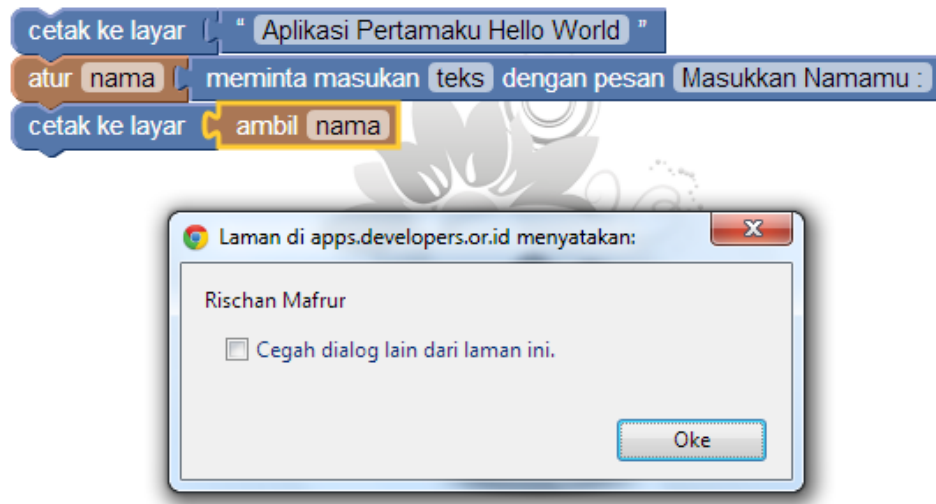
**Gambar 2.2 Index awal aplikasi IndoBlockly**

## 2. Input output

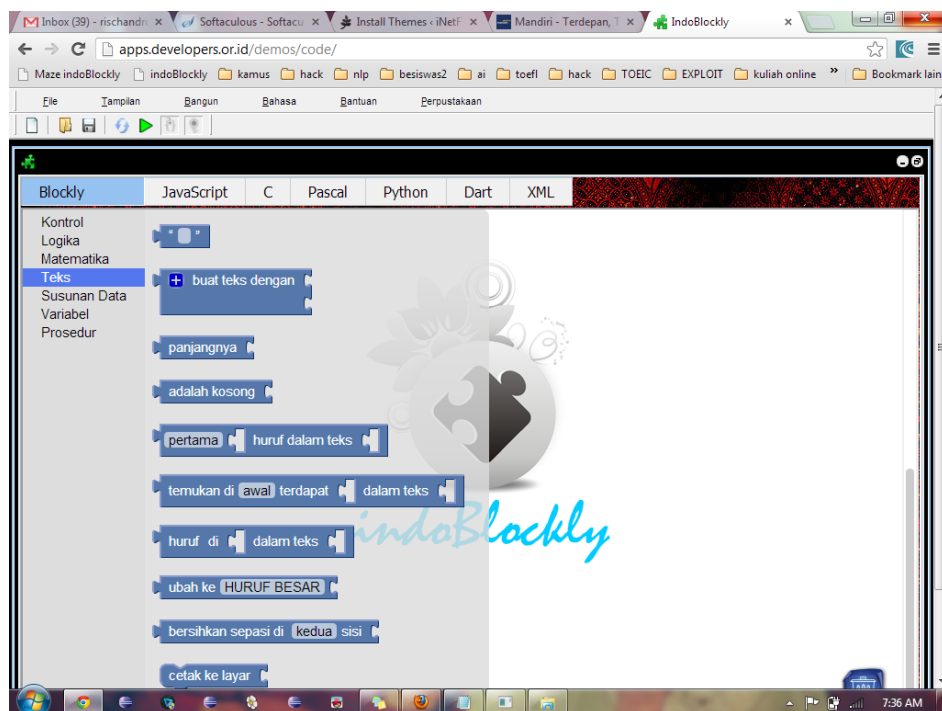
Pemrograman C menggunakan *scanf* untuk proses input dan *printf* untuk output sedangkan C++ menggunakan *cin* dan *cout*. Pada IndoBlockly user tinggal menekan menu text untuk mengambil puzzle cetak ke layar untuk output dan atur variabel kemudian meminta



masukkan teks/angka dengan pesan . Menggunakan input output pada IndoBlockly dapat dilihat pada Gambar 2.3 dan 2.4.



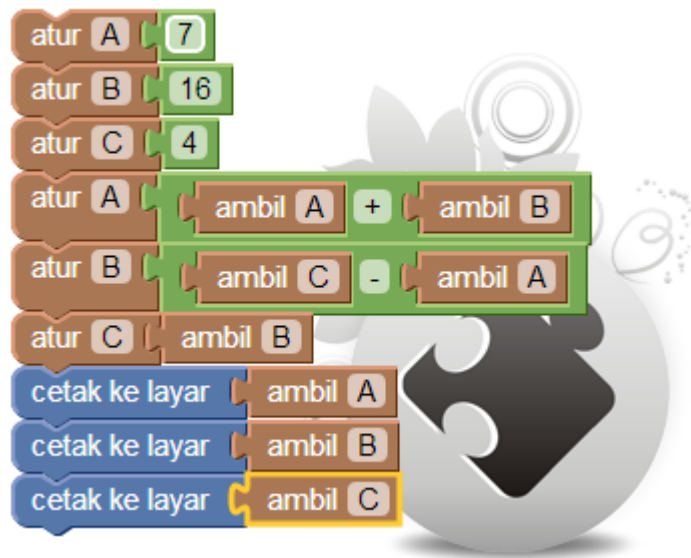
**Gambar 2.3** Input output sederhana menggunakan IndoBlockly



**Gambar 2.4** Menu Teks pada IndoBlockly

### 3. Variabel

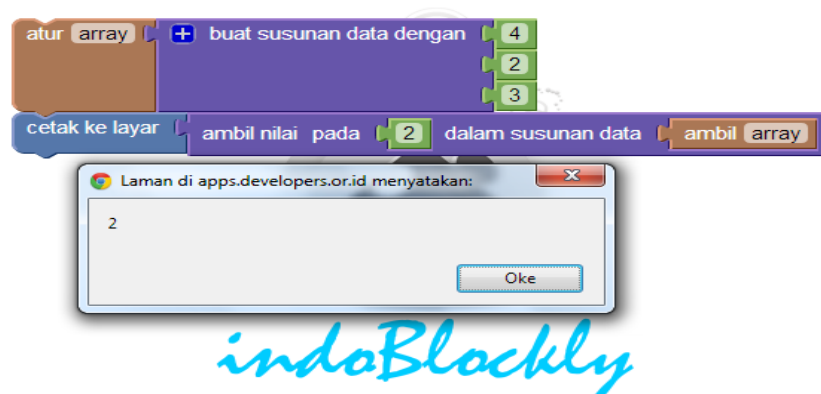
Pembuatan variabel pada IndoBlockly seperti pada pemrograman orientasi objek yaitu dengan menggunakan *set* dan *get* atau “atur” dan “ambil”. Tidak ada deklarasi seperti *int*, *string* dsb. Contoh penggunaan variabel dapat dilihat pada Gambar 2.5.



**Gambar 2.5 Contoh penggunaan variabel pada IndoBlockly**

### 4. Array

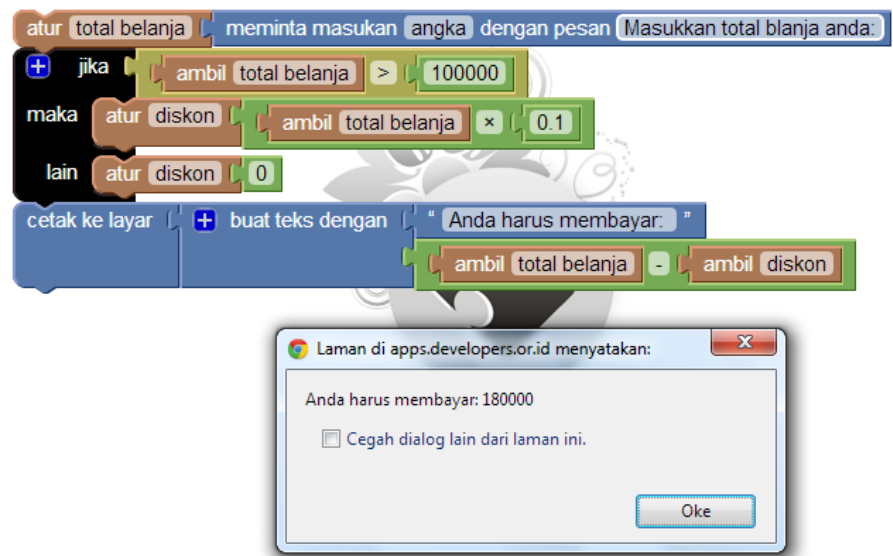
Gambar 2.6 adalah contoh penggunaan array menggunakan IndoBlockly.



**Gambar 2.6 Contoh array di IndoBlockly**

## 5. Kondisi

Biasanya pada beberapa bahasa pemrograman untuk sebuah kondisi itu menggunakan *if* atau *case*. IndoBlockly belum mendukung *case* jadi pada IndoBlockly hanya dapat menggunakan *if*. Contoh penggunaan *if* pada IndoBlockly dapat dilihat pada Gambar 2.7.



**Gambar 2.7 Penggunaan if pada IndoBlockly**

## 6. Perulangan

Perulangan pada IndoBlockly dapat menggunakan *for*, *while* dan *do while*. Gambar 2.8 adalah contoh penggunaan perulangan *for* menggunakan IndoBlockly.



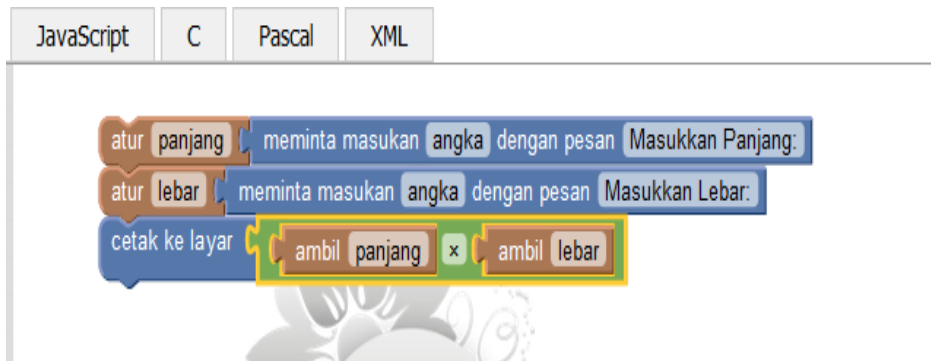
**Gambar 2.8 Penggunaan for pada IndoBlockly**

### 2.2.6.3 Contoh Pembuatan Program Sederhana menggunakan IndoBlockly

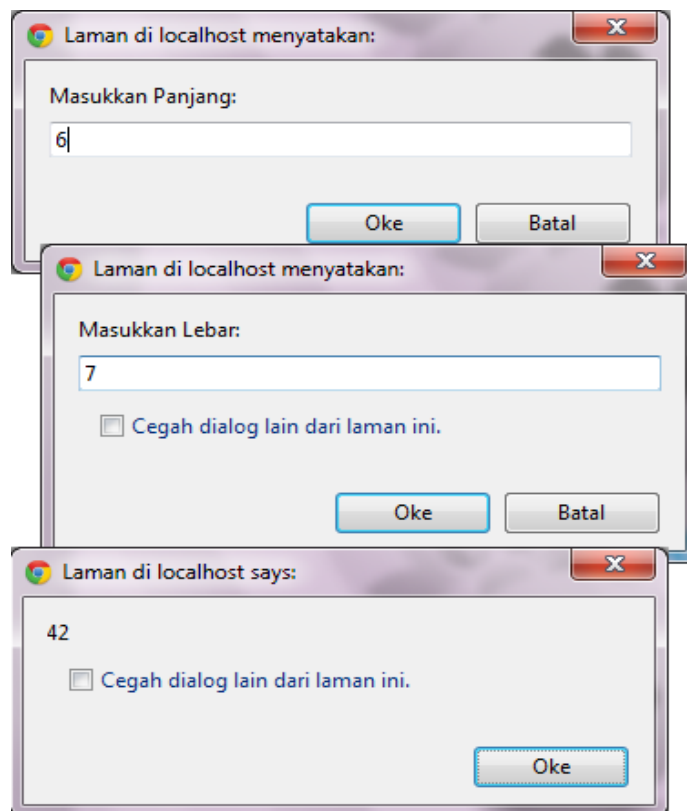
Pada pembahasan ini ada beberapa contoh program sederhana menggunakan IndoBlockly. Gambar 2.9 adalah contoh program untuk mencari nilai maksimum menggunakan IndoBlockly. Gambar 2.10 adalah program sederhana perhitungan luas persegi panjang. Fitur yang lain dari IndoBlockly adalah user dapat mengkonvert code puzzle IndoBlockly langsung menjadi source code C. Pada Gambar 2.11 adalah gambar hasil output aplikasi sederhana perhitungan persegi panjang menggunakan IndoBlockly dan Gambar 2.12 adalah gambar hasil konvert *source code* aplikasi persegi panjang menjadi *source code* C. Dan Gambar 2.13 adalah gambar hasil running program menggunakan C Free dari aplikasi sederhana perhitungan luas persegi panjang.



**Gambar 2.9 Program sederhana untuk mencari nilai maksimum dari inputan user menggunakan IndoBlockly**



**Gambar 2.10 Aplikasi sederhana perhitungan luas persegi panjang menggunakan IndoBlockly**



**Gambar 2.11 Hasil output running aplikasi sederhana perhitungan luas persegi panjang menggunakan IndoBlockly**

Blockly	JavaScript	<b>C</b>	Pascal	XML
---------	------------	----------	--------	-----

```

//-----STANDARD LIBRARY-----
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
//-----

#define maxChar 100

int panjang;
int lebar;

int main() {

    //input variabel Angka
    printf("Masukkan Panjang:");scanf("%d",&panjang);
    //input variabel Angka
    printf("Masukkan Lebar:");scanf("%d",&lebar);
    printf("%u ",panjang * lebar);

return 0;
}

```

**Gambar 2.12.** Hasil konvert code puzzle IndoBlockly ke *source code C*

The screenshot shows a C Free IDE window titled 'Untitled1.cpp'. The code is the same as in Gambar 2.12. Below the code editor, a terminal window titled '"D:\MASTER\cfree\C-Free Standard\temp\Untitled1.exe"' displays the program's output. The user has entered '6' for 'Masukkan Panjang:' and '7' for 'Masukkan Lebar:'. The program then outputs '42' followed by a space and 'Press any key to continue . . . \_'.

```

1 //-----STANDARD LIBRARY-----
2 #include <stdio.h>
3 #include <string.h>
4 #include <stdlib.h>
5 #include <math.h>
6 #include <ctype.h>
7 //-----
8 #define maxChar 100
9
10 int panjang;
11 int lebar;
12
13 int main() {
14
15     //input variabel Angka
16     printf("Masukkan Panjang:");scanf("%d",&panjang);
17     //input variabel Angka
18     printf("Masukkan Lebar:");scanf("%d",&lebar);
19     printf("%u ",panjang * lebar);
20
21 return 0;
22

```

"D:\MASTER\cfree\C-Free Standard\temp\Untitled1.exe"  
 Masukkan Panjang:6  
 Masukkan Lebar:7  
 42 Press any key to continue . . . \_

**Gambar 2.13** Hasil output running program dengan C Free.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Penelitian**

Metode penelitian adalah cara yang digunakan oleh peneliti dalam mengumpulkan data penelitian (Arikunto, 2007). Metode penelitian yang digunakan dalam penelitian ini adalah metode eksperimen. Metode eksperimen yaitu metode penelitian yang dipergunakan untuk meneliti suatu objek dengan melakukan suatu percobaan secara nyata di lapangan (Arikunto, 2006). Penelitian ini bersifat eksperimen, yaitu sengaja mengusahakan tumbuhnya variabel-variabel dan selanjutnya dikontrol untuk dilihat pengaruhnya terhadap kemampuan pemahaman konsep mahasiswa melalui hasil belajar mahasiswa. karena disini penulis membandingkan antara dua kelas yang menggunakan pendekatan konstruktivisme dan yang menggunakan pendekatan konvensional serta apakah pengaruhnya terhadap kemampuan pemahaman konsep mahasiswa melalui hasil belajar.

#### **3.2 Populasi dan Sampel**

Dibawah ini akan kami jelaskan mengenai populasi dan sampel dari penelitian kami, baik populasi dan sampel itu sendiri, teknik pengambilan sampel, design, alur, prosedur, persiapan, pelaksanaan dan penyelesaian penelitian.

### **3.2.1 Populasi**

Populasi adalah himpunan yang lengkap dari satuan-satuan atau individu yang karakteristiknya ingin diketahui (Anggoro, 2007). Populasi dalam penelitian ini adalah mahasiswa Teknik Informatika UIN Sunan Kalijaga Yogyakarta tahun 2012/2013 yang terdiri atas dua kelas yaitu kelas regular dan kelas mandiri.

### **3.2.2 Sampel**

Sampel adalah sebagian anggota populasi yang memberikan keterangan atau data yang diperlukan dalam suatu penelitian (Anggoro, 2007). Sampel yang digunakan pada penelitian ini menggunakan teknik *sampling* yaitu *purposive sampling*, sampelnya diambil berdasarkan tujuan tertentu. Sampel diambil dari populasi terjangkau yang dibagi menjadi dua kelompok. Kelompok pertama sebagai kelas eksperimen dan kelompok kedua sebagai kelas kontrol. Kedua kelompok tersebut diambil sebagai sampel karena memiliki karakteristik yang hampir sama.

### **3.3 Desain Penelitian**

Penelitian ini menggunakan desain *pretes postes control group design*, dalam desain ini digunakan dua kelompok subjek, satu diantaranya yang diberikan perlakuan. Desain penelitian ini dapat digambarkan sebagai berikut: (Emzir , 2008)



**Tabel 3.1 Desain Kelompok Eksperimen dan Kontrol Pretes-Posttes**

<i>Kelompok</i>	<i>Pretes</i>	<i>Treatment</i>	<i>Postes</i>
A	T <sub>1</sub>	X <sub>1</sub>	T <sub>2</sub>
B	T <sub>1</sub>	X <sub>2</sub>	T <sub>2</sub>

Keterangan:

A : Kelompok eksperimen (kelompok yang menggunakan *indoBlockly*)

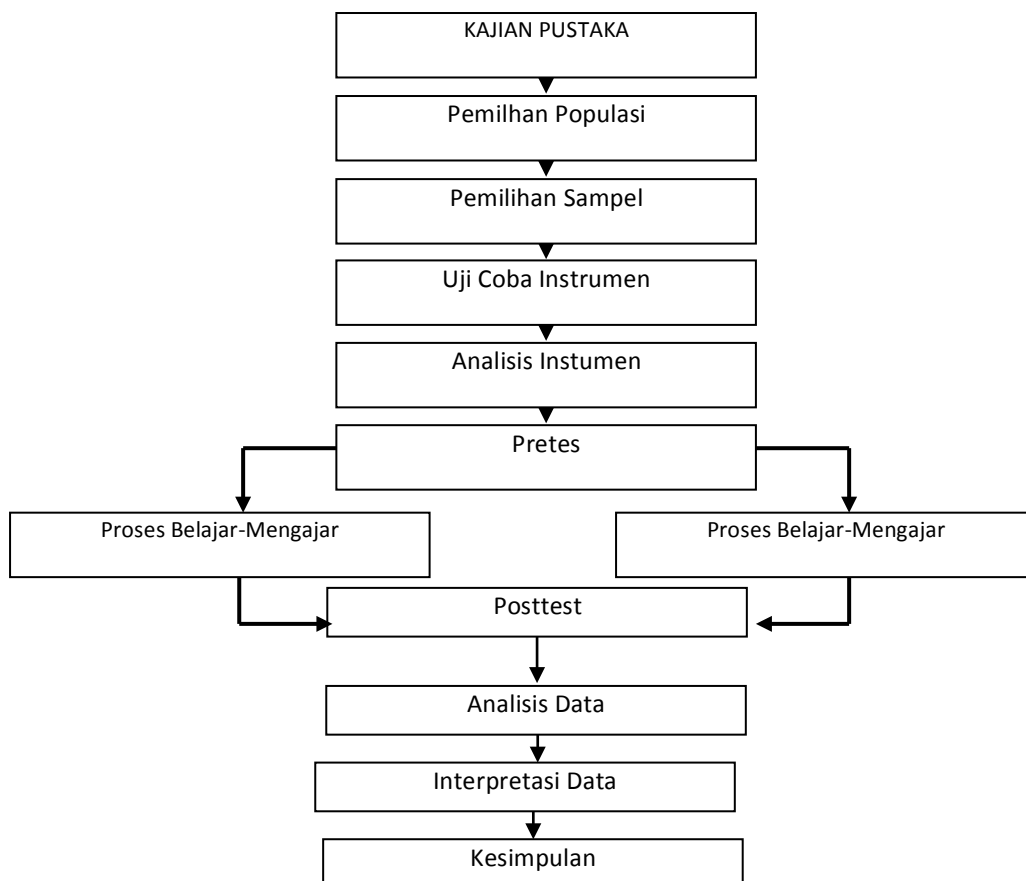
B : Kelompok kontrol (kelompok yang menggunakan model konvensional)

T<sub>1</sub> : Tes awal yang sama pada kedua kelompok (*prettes*)

T<sub>2</sub> : Tes akhir yang sama pada kedua kelompok (*posttes*)

X<sub>1</sub> : Perlakuan dengan menerapkan pembelajaran menggunakan *indoBlockly*

X<sub>2</sub> : Perlakuan dengan menerapkan model konvensional



**Gambar 3.1**

**Alur Penelitian**

### **3.4 Alur Penelitian**

Alur penelitian dapat dilihat pada diagram alur Gambar 3.1.

### **3.5 Prosedur Penelitian**

Untuk mempermudah pelaksanaan penelitian, maka perlu dirancang suatu prosedur penelitian yang berstruktur. Prosedur tersebut merupakan arahan dalam pelaksanaan penelitian dari awal sampai akhir, dengan harapan penelitian akan sesuai dengan yang telah direncanakan sebelumnya. Dalam penelitian ini prosedur penelitian dibagi dalam tiga tahap, yaitu tahap persiapan, pelaksanaan, dan penyelesaian penelitian.

#### **3.5.1 Persiapan Penelitian**

Dalam melaksanakan penelitian ini ditempuh melalui langkah-langkah sebagai berikut:

- a. Mengajukan judul penelitian.
- b. Merancang instrumen penelitian.
- c. Bersamaan dengan penyusunan instrumen, penulis memohon izin untuk melakukan penelitian pada mata kuliah pemrograman terstruktur teknik informatika UIN Sunan Kalijaga.
- d. Menyusun jadwal kegiatan penelitian.
- e. Menguji reliabilitas, validitas, daya pembeda serta indeks kesukaran instrumen uji coba, kemudian melakukan revisi.
- f. Pelaksanaan penelitian.

### **3.5.2 Pelaksanaan Penelitian**

Langkah-langkah yang dilakukan pada pelaksanaan penelitian adalah:

- a. Memilih sampel *purposive sampling* sebanyak dua kelas.
- b. Memberikan lembar soal pretes yang sama kepada kedua kelompok sampel kelas tersebut.
- c. Proses pembelajaran pada kelas eksperimen dengan menggunakan IndoBlockly dan kelas kontrol dengan menggunakan model praktikum konvensional.
- d. Setelah proses pembelajaran berakhir, dilakukan postes untuk kedua kelompok yang diteliti.
- e. Diadakan penilaian hasil pretes dan postes pada kedua kelompok kelas tersebut.

Langkah-langkah pelaksanaan penelitian dapat dilihat pada Lampiran A Pelaksanaan Penelitian halaman 79.

### **3.5.3 Penyelesaian Penelitian**

Langkah-langkah penyelesaian penelitian adalah sebagai berikut:

- a. Menganalisis data hasil pretes dan postes dari kelompok eksperimen dan kelompok kontrol.
- b. Mengucapkan terima kasih kepada Bapak Sumarsono, M.Kom selaku dosen pemrograman terstruktur yang telah mengizinkan dan penulis dalam melaksanakan penelitian.
- c. Penyusunan skripsi.

### **3.6 Instrumen Penelitian**

Instrumen penelitian merupakan alat bantu untuk pengumpulan dan pengolahan data tentang variabel-variabel yang diteliti. Oleh karena itu untuk memperoleh data dalam menjawab permasalahan untuk menguji hipotesis yang telah dirumuskan, maka diperlukan alat atau instrumen. Adapun instrumen dalam penelitian ini adalah:

#### **3.6.1 Seperangkat Soal**

Seperangkat soal dalam penelitian ini berupa tes, dimana untuk mengukur ada atau tidaknya serta besarnya pengaruh model pembelajaran menggunakan IndoBlockly terhadap pemahaman konsep algoritma pada mata kuliah pemrograman terstruktur.

Tes ini dikenakan kepada kedua kelompok subyek penelitian dengan kriteria tes yang sama, yaitu tes awal (pretes) dan tes akhir (postes). Tes awal dan tes akhir ini diadakan untuk mengetahui tingkat pemahaman algoritma sebelum dan sesudah pemberian perlakuan.

Bentuk tes yang digunakan adalah tes uraian sebanyak sepuluh soal. Soal tes sebelumnya diujicobakan terlebih dahulu untuk mengetahui kualitas instrumen atau alat pengumpul data yang digunakan. Instrumen disebut berkualitas dan dapat dipertanggungjawabkan pemakaiannya apabila sudah terbukti validitas, reliabilitas, indeks kesukaran, dan daya pembedanya.

### 3.6.1.1 Menentukan Validitas Soal

Suatu alat evaluasi disebut valid (absah atau sah) apabila alat tersebut mampu mengevaluasi apa yang seharusnya di evaluasi. Validitas butir soal essay (uraian) dihitung dengan menggunakan rumus koefisien korelasi *Product Moment* memakai angka kasar, yaitu (Suherman dan Sukjaya, 1990):

$$r_{xy} = \frac{N \cdot \sum XY - \sum X \sum Y}{\sqrt{(N \cdot \sum X^2 - (\sum X)^2)(N \cdot \sum Y^2 - (\sum Y)^2)}}$$

Keterangan:

$r_{xy}$  = Koefisien korelasi antara variebel  $X$  dan  $Y$

$N$  = banyaknya mahasiswa yang mengikuti tes

$X$  = nilai hasil uji coba

$Y$  = skor total

**Tabel 3.2**

#### **Klasifikasi Koefisien Validitas**

No	Nilai $r_{xy}$	Interpretasi
1	$0,80 < r_{xy} \leq 1,00$	Validitas Sangat Tinggi
2	$0,60 < r_{xy} \leq 0,80$	Validitas Tinggi
3	$0,40 < r_{xy} \leq 0,60$	Validitas Sedang
4	$0,20 < r_{xy} \leq 0,40$	Validitas Rendah
5	$0,00 < r_{xy} \leq 0,20$	Validitas Sangat Rendah
6	$r_{xy} \leq 0,00$	Tidak Valid

Sumber: (Suherman dan Sukjaya, 1990)

Kemudian untuk menguji keberartian validitas (koefisien korelasi) soal essay digunakan statistik uji  $t$  yang dikemukakan oleh (Sugiyono, 2007) yaitu:

$$t = r_{xy} \sqrt{\frac{n-2}{1-r_{xy}^2}}$$

Keterangan:  $t$  = daya beda

Bila  $t_{hitung} > t_{tabel}$  maka soal sah tetapi jika  $t_{hitung} \leq t_{tabel}$ , maka soal tersebut tidak sah dan tidak digunakan untuk instrumen penelitian.

### 3.6.1.2 Menentukan Reabilitas Soal

Reliabilitas suatu alat ukur sebagai suatu alat yang memberikan hasil yang tetap sama (konsisten, ajeg). Hasil pengukuran itu harus tetap sama (relative sama) jika pengukurannya diberikan kepada subjek yang sama meskipun dilakukan oleh orang yang berbeda, waktu berbeda, dan tempat yang berbeda. Menurut Suherman dan Sukjaya untuk menentukan reliabilitas soal berbentuk essay (uraian) digunakan rumus *Alpha*, yaitu:

$$r_{11} = \left( \frac{n}{n-1} \right) \left( 1 - \frac{\sum S_i^2}{S^2} \right)$$

Keterangan:

$r_{11}$  = koefisien reliabilitas instrumen

$n$  = banyaknya butir soal

$\sum S_i^2$  = jumlah skor tiap butir soal

$S^2$  = varians skor total

Sedangkan untuk menghitung varians skor digunakan rumus:

$$S_i^2 = \frac{\sum x_i^2 - \frac{(\sum x_i)^2}{N}}{N}$$

*Keterangan:*

$N$  = banyaknya sampel / peserta tes

$x_i$  = skor butir soal ke- $i$

$i$  = nomor soal

**Tabel 3.3**

**Klasifikasi Koefisien Reliabilitas**

No	Nilai $r_{11}$	Interpretasi
1	$0,90 < r_{11} \leq 1,00$	Sangat tinggi
2	$0,70 < r_{11} \leq 0,90$	Tinggi
3	$0,40 < r_{11} \leq 0,70$	Sedang
4	$0,20 < r_{11} \leq 0,40$	Rendah
5	$r_{11} \leq 0,20$	Sangat rendah

Sumber: (Suherman dan Sukjaya, 1990)

### 3.6.1.3 Menentukan Daya Pembeda Soal

Seberapa jauh kemampuan butir soal tersebut mampu membedakan antara testi yang mengetahui jawabannya dengan benar dengan testi yang tidak dapat menjawab soal tersebut (Suherman dan Sukjaya, 1990). Daya pembeda dihitung dengan menggunakan rumus DP untuk tes sebagai berikut:

$$DP = \frac{SA - SB}{\frac{1}{2} \times N \times Maks}$$

Keterangan:

$DP$  = daya pembeda

$SA$  = jumlah skor yang dicapai mahasiswa kelompok atas

$SB$  = jumlah skor yang dicapai mahasiswa kelompok bawah

$N$  = jumlah mahasiswa dari kelompok atas dan kelompok bawah

$Maks$  = Skor maksimal

**Tabel 3.4**

**Klasifikasi Koefisien Daya Pembeda**

No	Nilai Daya Pembeda ( $DP$ )	Interpretasi
1	$DP \leq 0,00$	Sangat jelek
2	$0,00 < DP \leq 0,20$	Jelek
3	$0,20 < DP \leq 0,40$	Cukup
4	$0,40 < DP \leq 0,70$	Baik
5	$0,70 < DP \leq 1,00$	Sangat Baik

Sumber: Suherman dan Sukjaya (1990)

### 3.6.1.4 Menentukan Tingkat Kesukaran Soal

Untuk menghitung tingkat kesukaran soal yang berbentuk uraian digunakan rumus:

$$IK = \frac{SA + SB}{N \times Maks}$$

Keterangan:

$IK$  = Indeks kesukaran tiap butir soal

$SA$  = Jumlah skor yang dicapai mahasiswa kelompok atas

$SB$  = jumlah skor yang dicapai mahasiswa kelompok bawah



$N$  = Jumlah mahasiswa dari kelompok atas dan kelompok bawah

$Maks$  = skor maksimal

**Tabel 3.5**

**Klasifikasi Koefisien Indeks Kesukaran**

No	Nilai Daya Pembeda ( $DP$ )	Interpretasi
1	$IK \leq 0,00$	Terlalu Sukar
2	$0,00 < IK \leq 0,30$	Sukar
3	$0,30 < IK \leq 0,70$	Sedang
4	$0,70 < IK \leq 1,00$	Mudah
5	$IK = 1,00$	Sangat Mudah

Sumber: (Suherman dan Sukjaya, 1990)

### 3.6.2 Hasil Uji Instrumen

Hasil uji instrumen dibahas pada bab III ini dikarenakan uji instrumen tidak termasuk dalam pembahasan. Instrumen penelitian merupakan alat bantu untuk pengumpulan dan pengolahan data tentang variabel-variabel yang diteliti. Dalam kasus ini instrumen penelitian adalah soal. Uji instrumen perlu dilakukan karena untuk menguji apakah instrumen itu layak atau tidak digunakan dalam penelitian. Jika hasil uji instrumen mengatakan sebuah instrumen itu tidak layak digunakan maka jelas instrumen tersebut tidak boleh digunakan dalam penelitian. Ada dua cara uji instrumen dan dalam penelitian ini kedua-duanya sudah dilakukan yaitu uji instrumen expert yaitu dengan bertanya dengan ahlinya dalam kasus ini yaitu dosen pemrograman terstruktur. Kedua yaitu uji instrumen uji coba dengan cara mengujikan soal

tersebut kepada sampel kemudian hasil skornya dianalisis menggunakan analisis uji statistik uji instrumen.

#### **3.6.2.1 Hasil Uji Instrumen Expert**

Sesuai dengan batasan masalah bahwa penelitian ini hanya menguji pada pemahaman mengenai input output, variabel, array, kondisi, dan perulangan. Penulis sudah membuat soal yang terdiri dari 10 soal uraian yang masing-masing soal mempunyai skor 10 sehingga total skor adalah 100. Dari 10 soal yang penulis ajukan pada dosen pemrograman terstruktur semuanya diterima dan dinyatakan telah sesuai dengan tiga soal yang perlu di revisi. Revisi tiga soal tersebut sebagai berikut :

1. Soal nomor satu mengenai flowchart ditambah soal untuk menyebutkan jenis-jenis flowchart yang mahasiswa kenal.
2. Soal nomor tiga mengenai array ditambah array dua dimensi jadi tidak hanya array satu dimensi.
3. Soal nomor delapan mengenai mencari bilangan terbesar direvisi menjadi soal dengan variabel angka statis tidak menggunakan input angka dinamis.

#### **3.6.2.2 Hasil Uji Instrumen Uji Coba**

Teknik pengolahan data hasil dari uji instrumen menggunakan *MS Excel* dengan langsung mengimplementasikan rumus uji validitas, reabilitas, daya beda dan tingkat kesukaran. Hasil perhitungan uji coba instrumen dapat dilihat pada Tabel 3.6. Dari hasil analisis uji coba instrumen seluruh soal

memenuhi kriteria untuk dipakai dalam penelitian. Akan tetapi ada tiga soal yaitu soal nomor 8,9, dan 10 yang mempunyai daya beda jelek dan tergolong sulit, itu terbukti dari hasil test uji coba instrumen hanya ada satu mahasiswa yang mendapatkan skor cukup bagus untuk soal nomor 8,9, dan 10. Sehingga berdasarkan hasil penelitian uji instrument penulis memutuskan untuk menggunakan tujuh soal yaitu soal nomor 1,2,3,4,5,6, dan 7 dalam pretes dan postes. Perhitungan selengkapnya mengenai uji coba instrument ini dapat dilihat pada Lampiran B Analisis Uji Instrumen halaman 80.

Soal pretes dan postes yang telah direvisi berdasarkan hasil uji instrumen baik uji instrumen expert maupun uji instrumen uji coba soal dapat dilihat pada Lampiran C Soal Pretes dan Postes halaman 82.