

**KORELASI MODEL PEMBELAJARAN MENGGUNAKAN
INDOBLOCKLY TERHADAP PEMAHAMAN MAHASISWA
PADA MATA KULIAH PEMROGRAMAN TERSTRUKTUR**

RESUME SKRIPSI

untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1

Program Studi Teknik Informatika



Disusun Oleh

Rischan Mafrur

09650007

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UIN SUNAN KALIJAGA
YOGYAKARTA**

2013

KORELASI MODEL PEMBELAJARAN MENGGUNAKAN INDOBLOCKLY TERHADAP PEMAHAMAN MAHASISWA PADA MATA KULIAH PEMROGRAMAN TERSTRUKTUR

INTISARI

Kemajuan teknologi dan industri baik di ranah nasional maupun internasional menyebabkan meningkatnya kebutuhan terhadap SDM ahli 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 *programming environment* adalah *teaching system*. Bagaimana dapat membentuk SDM yang ahli jika sistem pembelajaran tidak mendukung ? Oleh karena itu peneliti mengambil studi di kampus UIN Sunan Kalijaga Yogyakarta pada mata kuliah pemrograman terstruktur tahun 2012/2013. Pembelajaran pada mata kuliah pemrograman terstruktur adalah menggunakan *C Free*. Peneliti mengusulkan penggunaan 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 dijumpai oleh programmer pemula.

Penelitian ini lebih ditekankan pada hubungan penggunaan IndoBlockly terhadap hasil belajar mahasiswa pada mata kuliah pemrograman terstruktur. Analisis yang digunakan meliputi analisis deskriptif, perhitungan index gain, analisis inferensi dan analisis korelasi. Data yang kami analisis adalah data nilai mahasiswa hasil dari pretes dan postes. Hasil dari penelitian ini adalah berupa kesimpulan apakah penggunaan IndoBlockly mempunyai pengaruh terhadap hasil pembelajaran mahasiswa pada mata kuliah pemrograman terstruktur.

Analisis uji hitpotesis beda rata-rata untuk data pretes menghasilkan *p-value* $0,749 > 0,05$ sehingga H_0 diterima yaitu tidak ada perbedaan rata-rata pretes kelas eksperimen dan kelas kontrol. Ini menjadi bukti bahwa kelas kontrol dan kelas eksperimen keduanya homogen. Sebaliknya analisis beda rata-rata postes kelas eksperimen dan kelas kontrol menghasilkan *p-value* $0,000 < 0,05$, H_0 ditolak kesimpulannya adalah rata-rata kelas eksperimen dan kelas kontrol berbeda. Perhitungan gain diperoleh rata-rata gain kelas eksperimen sebesar 0,63 adalah gain sedang dan kelas kontrol 0,16 adalah gain rendah, ini menunjukkan bahwa nilai gain kelas eksperimen lebih besar dari kelas kontrol. Analisis korelasi menghasilkan koefisien korelasi sebesar 0,43. Kemudian dilakukan uji signifikansi koefisien korelasi diperoleh *p-value* $0,022 < 0,05$, H_0 ditolak dan disimpulkan bahwa kontribusi variabel independen (X) yaitu penggunaan IndoBlockly terhadap variabel dependen (Y) yaitu hasil belajar mahasiswa adalah signifikan. Hasil perhitungan determinasi yaitu r^2 sebesar 0,1849 atau 18,49 % menunjukkan bahwa IndoBlockly memberikan pengaruh 18,49 % terhadap peningkatan hasil belajar pemrograman terstruktur. Jadi model pembelajaran menggunakan IndoBlockly terbukti lebih baik dibandingkan dengan model pembelajaran konvensional dengan pengaruh sebesar 18,49 % terhadap peningkatan hasil belajar.

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

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.

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 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 (Wislow, 1996). Sistem pendidikan di Negara maju sudah mengantisipasi hal tersebut, bahasa pemrograman sudah mulai dikenalkan kepada anak-anak mulai dari 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 tersebut di-*design* dengan tujuan menjadikan belajar memprogram itu menyenangkan, disamping itu software tersebut juga dikemas dengan begitu menarik dan pengguna hanya perlu menyusun *puzzle* untuk membuat sebuah aplikasi atau program. Software-software tersebut tidak hanya di peruntukkan untuk anak-anak TK dan SD, tapi diperuntukkan bagi siapa saja yang masih pemula dalam pemrograman. (Mafrur, 2012).

Pada studi 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 dapat 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).

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 ini sebagai berikut:

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 dapat mempengaruhi kedua variabel tersebut.
3. Peneliti tidak menghitung besaran faktor eksternal yang peneliti sebutkan pada point ke 2.
4. Proses pembelajaran dilakukan lima kali pertemuan dengan penekanan pada pemahaman input output, variabel, array, kondisi, dan perulangan.

1.4 Tujuan Penelitian

Mengetahui apakah pembelajaran menggunakan *IndoBlockly* dapat meningkatkan pemahaman (hasil belajar) 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 dapat 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. Begitu juga untuk penelitian implementasi dari sebuah tool kemudian menganalisis apakah tool tersebut dapat membantu pemahaman dalam mempelajari suatu hal, penelitian semacam ini juga sudah banyak dilakukan oleh berbagai peneliti. Akan tetapi tool yang kami teliti adalah IndoBlockly, sebuah tool 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

1.1 Tinjauan Pustaka

No	Peneliti	Judul Penelitian	Hasil	Tahun
1	Robins, A, J. Rountree, and N. Rountree	Learning and teaching programming: A review and discussion	Penelitian ini menghasilkan kesimpulan bahwa faktor yang mempengaruhi proses dari programmer pemula menjadi programmer ahli tidak hanya faktor dari dalam diri programmer tetapi juga ada faktor eksternal yang tidak kalah penting yaitu strategi belajar(belajar mengajar).	2003
2	Lahtinen, E., Mutka, K. A., and Jarvinen, H. M	A Study of the difficulties of novice programmers	Penelitian ini membahas mengenai beberapa kesulitan yang sering dijumpai oleh programmer pemula meliputi masalah-masalah teknis dan juga ternyata masalah sosial programmer itu juga menjadi faktor yang perlu diperhatikan.	2005
3	Kelleher dan Pausch	Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers	Penelitian ini mengelompokkan <i>programming environment</i> menjadi beberapa taksonomi, dan taksonomi tertinggi adalah <i>learning system</i> .	2003

4	Maloney, J., Peppler, K., Kafai, Y., Resnick, M., and Rusk, N	Programming by Choice: Urban Youth Learning Programming with Scratch	Penelitian ini yang sangat menginspirasi penulis. Penelitian ini dilakukan oleh tim dari Scratch dengan dukungan dana penuh dari MIT. MIT membangun sebuah Scratch club house kemudian selama 18 bulan penelitian dilakukan, objek penelitiannya adalah orang—rang/anak- anak dan proyek- proyek/aplikasi yang dihasilkan dengan menggunakan scratch di scratch club house tersebut.	2008
---	---	--	---	------

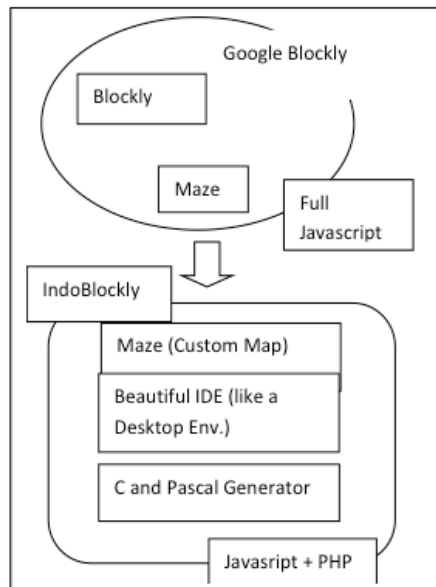
1.2 Landasan Teori

Dalam landasan teori ini menjelaskan mengenai Indoblockly yaitu tool yang produk/tool yang menjadi objek dari penelitian ini.

1.2.1 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).

1.2.2 Konsep IndoBlockly



Gambar 2.1 . Konsep IndoBlockly

Konsep IndoBlockly adalah membuat sebuah software dengan kriteria sebagai berikut :

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 dapat dikonvert ke dalam source code C atau Pascal dan langsung dapat di eksekusi dengan C atau Pascal compailer
7. Format file penyimpanan yang portabel.
8. Ada beberapa game logika untuk mengasah kemampuan otak anak.

9. Dapat 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 Blockly adalah blok-blok puzzle dari *Google Blockly*. Konsep IndoBlockly dalam bentuk gambar dapat dilihat pada Gambar 2.2.

1.2.3 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 screenshot dari aplikasi indoBlockly.

1. Halaman Indeks IndoBlockly

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



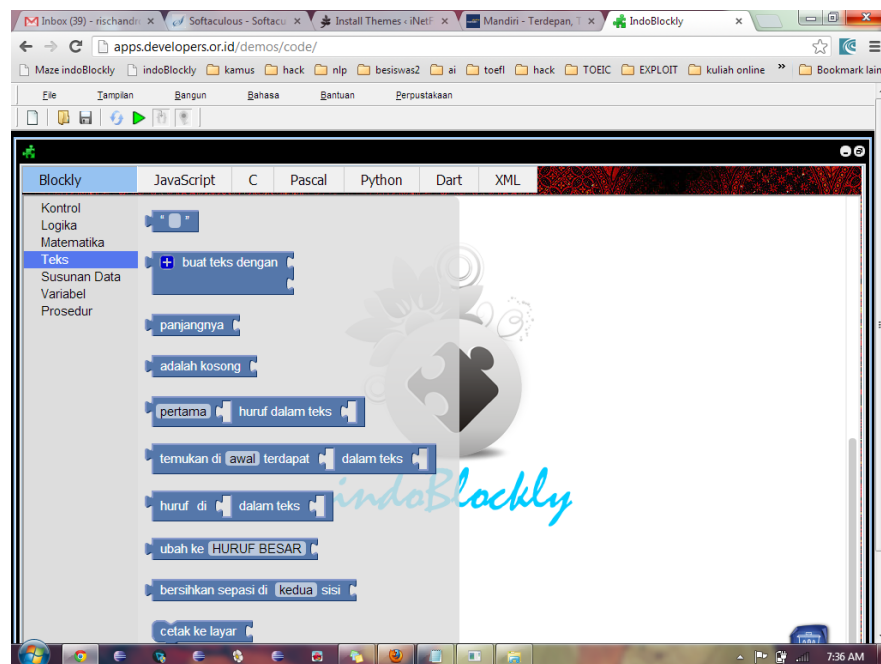
Gambar 2.2 Halaman Indeks aplikasi IndoBlockly

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 dengan pesan . Penggunaan input output pada IndoBlockly dapat dilihat pada Gambar 2.3 dan 2.4.



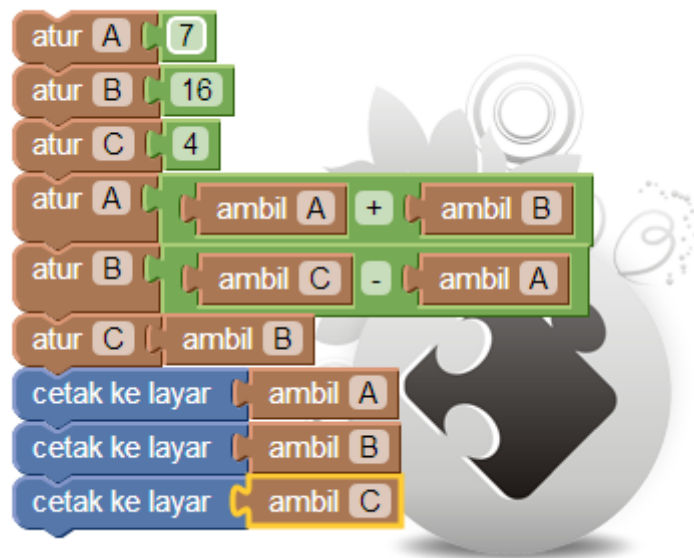
Gambar 2.3 Input output sederhana menggunakan IndoBlockly



Gambar 2.4 Menu Teks pada IndoBlockly

3. Variabel

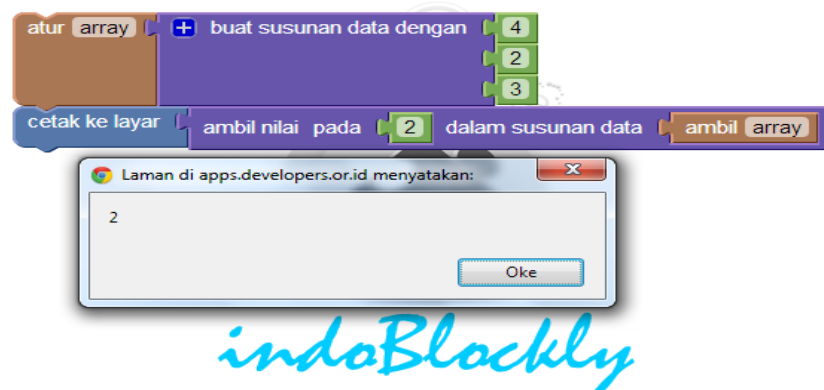
Pembuatan variabel pada IndoBlockly seperti pada pemrograman orientasi objek yaitu dengan menggunakan *set* dan *get* atau “atur” dan “ambil”. Tidak ada deklarasi seperti *int*, *string* dsb. Contoh penggunaan variable dapat dilihat pada Gambar 2.5.



Gambar 2.5 Contoh penggunaan variabel pada IndoBlockly

4. Array

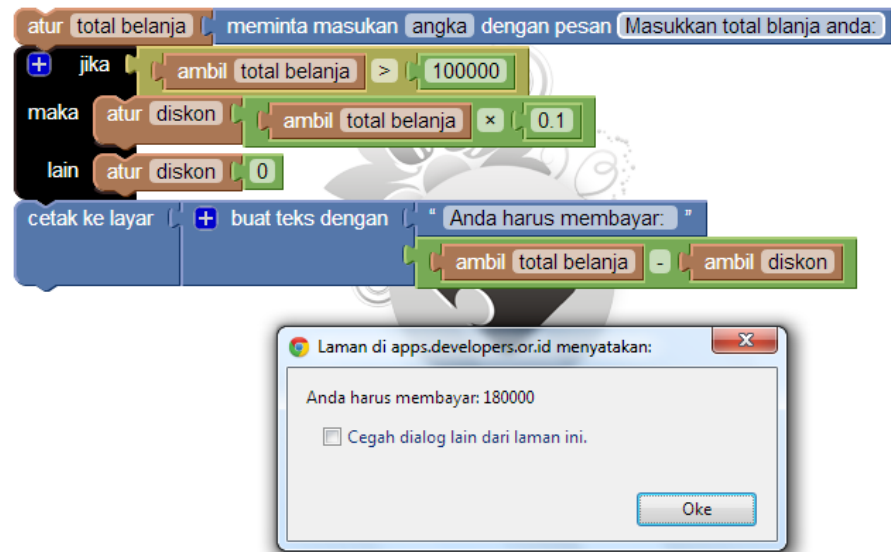
Gambar 2.6 adalah contoh penggunaan array menggunakan IndoBlockly.



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.



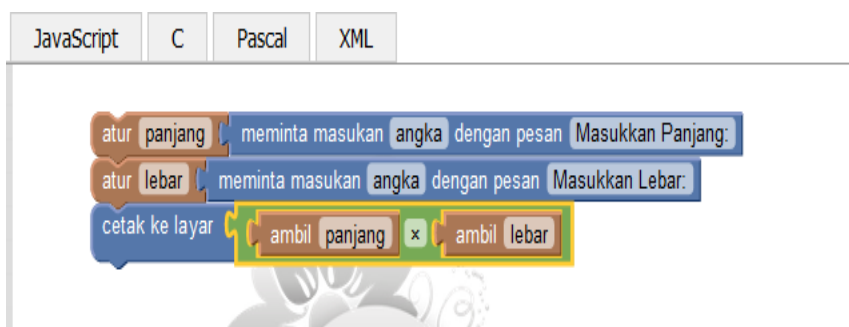
Gambar 2.8 Penggunaan *for* pada IndoBlockly

1.2.4 Contoh Pembuatan Program Sederhana menggunakan IndoBlockly

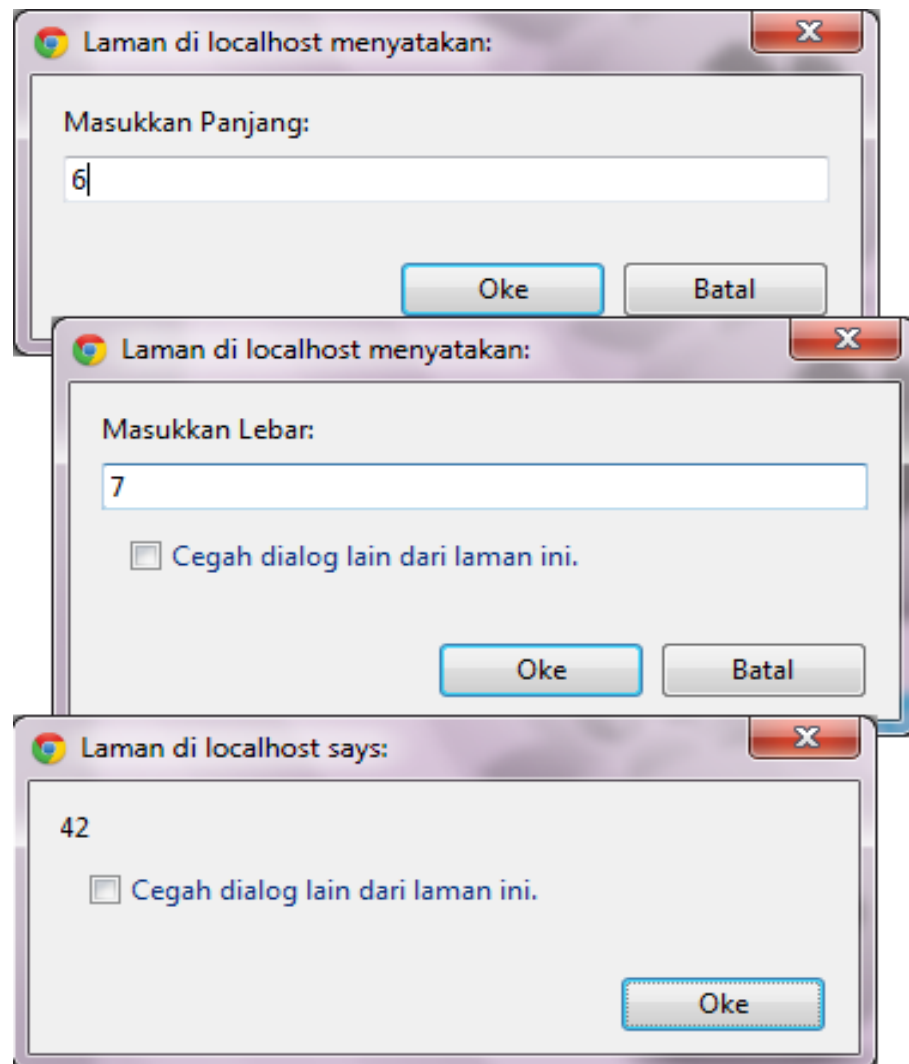
Pada pembahasan ini ada beberapa contoh program sederhana menggunakan IndoBlockly. Gambar 2.9 adalah contoh program untuk mencari nilai maksimum menggunakan IndoBlockly. Gambar 2.10 adalah program sederhana perhitungan luas persegi panjang. Fitur yang lain dari IndoBlockly adalah user dapat mengkonvert 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.



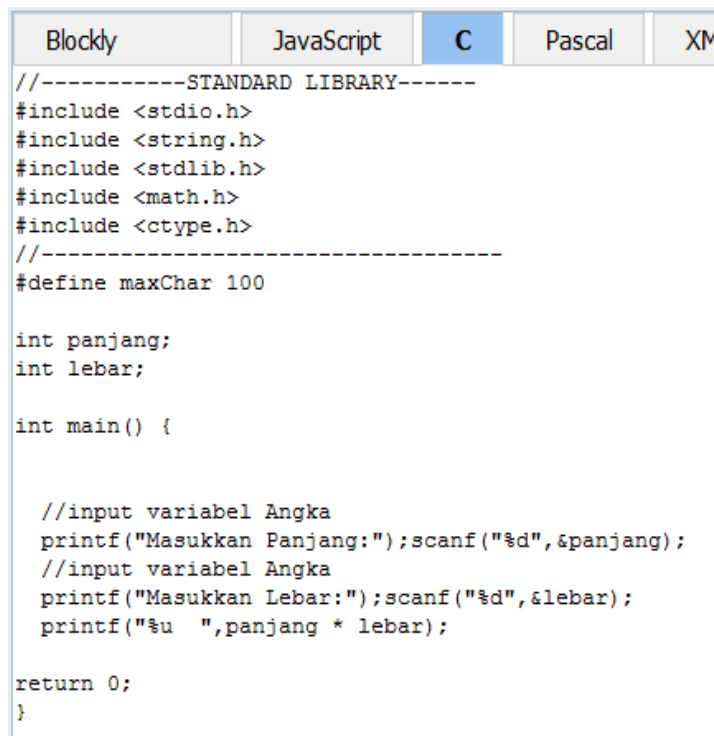
Gambar 2.9 Program sederhana untuk mencari nilai maksimum dari inputan user menggunakan IndoBlockly



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



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

#define maxChar 100

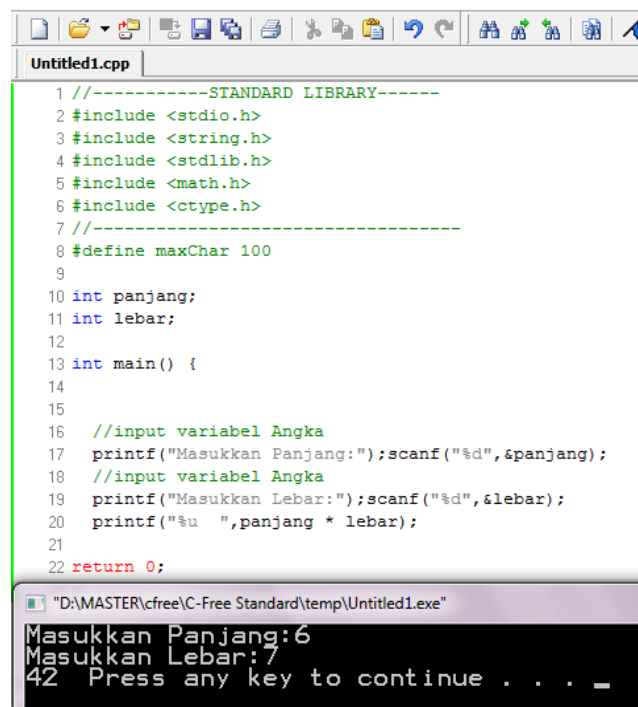
int panjang;
int lebar;

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*



```
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
16     //input variabel Angka
17     printf("Masukkan Panjang:");scanf("%d",&panjang);
18     //input variabel Angka
19     printf("Masukkan Lebar:");scanf("%d",&lebar);
20     printf("%u ",panjang * lebar);
21
22     return 0;
}
```

"D:\MASTER\cfree\C-Free Standard\temp\Untitled1.exe"

Masukkan Panjang:6
Masukkan Lebar:7
42 Press any key to continue . . . _

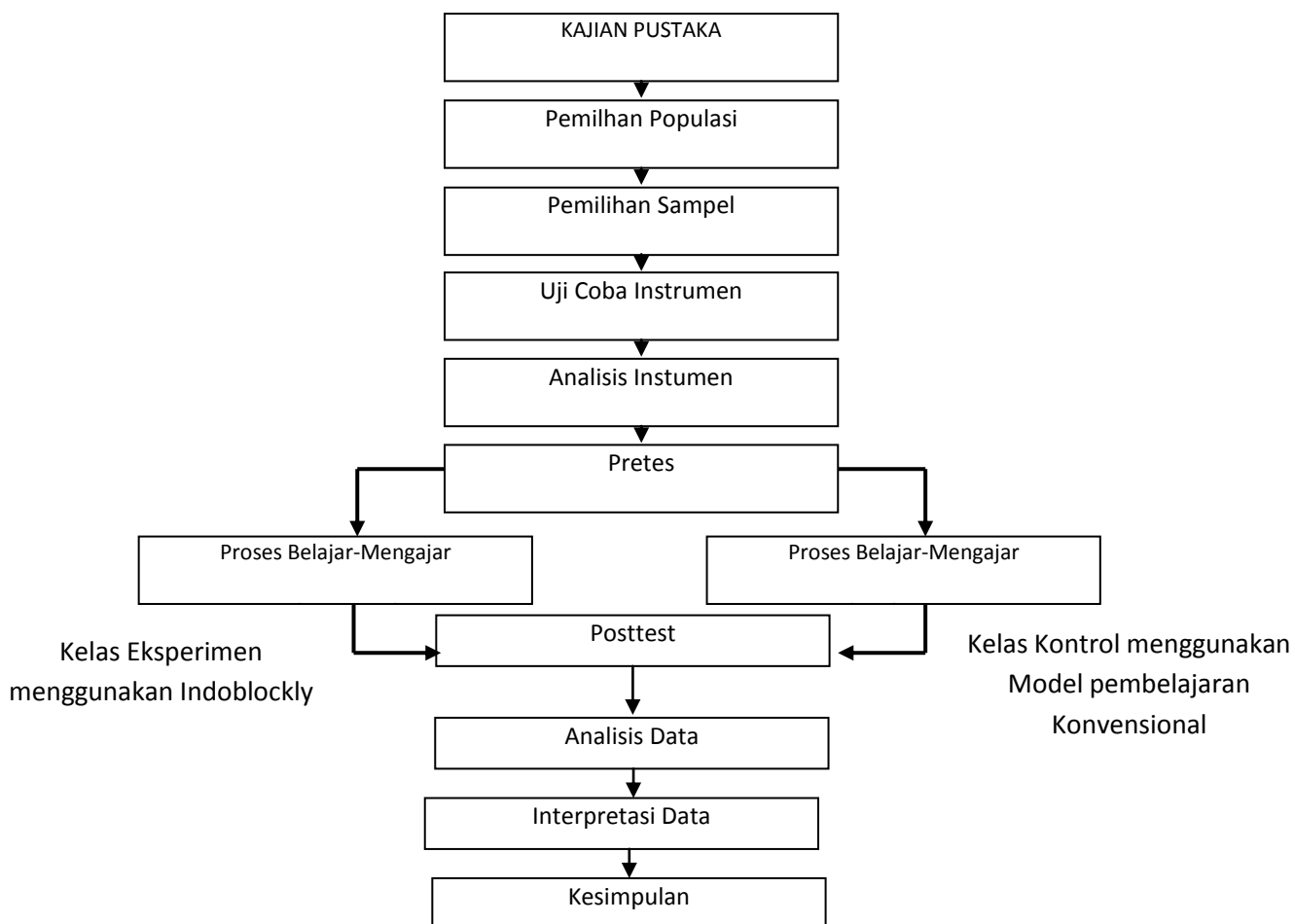
Gambar 2.13 Hasil output running program dengan C Free.

BAB III

METODE PENELITIAN

3.1 Alur dan Desain Penelitian

Penelitian ini menggunakan desain *pretes postes control group design*, dalam desain ini digunakan dua kelompok subjek, satu diantaranya yang diberikan perlakuan. Alur penelitian ini dapat digambarkan pada Gambar 3.1 sebagai berikut: (Emzir , 2008).



Gambar 3.1

Alur Penelitian

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Hasil Penelitian dan Pembahasan

Soal pretes dan postes yang tadinya berjumlah sepuluh butir soal kemudian dilakukan uji instrumen expert dan analisis uji coba soal menghasilkan tujuh soal yang valid sehingga hanya tujuh soal yang digunakan untuk pretes dan postes. Populasi dari penelitian ini adalah mahasiswa baru Teknik Informatika UIN Sunan Kalijaga 2012/2013 yang berjumlah kurang lebih seratus mahasiswa yang terdiri dari dua kelas yaitu kelas regular dan kelas mandiri. Sampel pada penelitian ini adalah kelas regular yang berjumlah 56 mahasiswa kemudian kami bagi menjadi dua kelas yaitu kelas kontrol dan kelas eksperimen. Pengambilan sampel berdasarkan teori *purposive sampling* dengan jumlah sampel 28 kelas kontrol dan 28 kelas eksperimen. Jumlah sampel tersebut sudah memenuhi kaidah pengambilan sampel sesuai dengan teori Gay dan Diehl.

4.1.1 Analisis Deskriptif Data Hasil Pretes dan Postes

Tabel 4.1 Statistik Deskriptif Data Hasil Pretes dan Postes

Kelas	Banyak Mhs	Pretes				Postes			
		Min	Max	Mean	Sd	Min	Max	Mean	Sd
Eksperimen	28	5,00	65,00	19,32	12,46	37,00	70,00	50,61	10,54
Kontrol	28	7,00	50,00	19,25	9,78	5,00	67,00	26,39	15,09

Berdasarkan Tabel 4.1 dapat dilihat bahwa mahasiswa pada kelas eksperimen dan kelas kontrol berjumlah sama, yaitu masing-masing kelas terdiri dari 28 mahasiswa. Rata-rata skor pretes kelas eksperimen dan kelas kontrol masing-masing adalah 19,32 dan 19,25 dimana rata-rata skor pretes kelas eksperimen lebih tinggi sedikit (hanya terpaut koma) daripada rata-rata skor pretes kelas kontrol. Sedangkan rata-rata skor postes kelas eksperimen dan kelas kontrol masing-masing adalah 50,61 dan 26,39 dari rata-rata ini juga sudah terlihat bahwa rata-rata skor postes kelas eksperimen yang menggunakan

IndoBlockly lebih tinggi daripada skor kelas kontrol yang menggunakan pembelajaran model konvensional. Sementara itu, standar deviasi pretes kelas eksperimen dan kelas kontrol masing-masing adalah 12,46 dan 9,78 yaitu kelas eksperimen lebih besar standar deviasinya dibandingkan kelas kontrol. Akan tetapi, untuk standar deviasi postes dimana kelas kontrol lebih besar daripada kelas eksperimen masing-masing adalah 15,09 dan 10,54.

4.1.2 Analisis Inferensi Data Hasil Pretes dan Postes

Analisis inferensi dilakukan terhadap data pretes dan postes baik kelas eksperimen maupun kelas kontrol. Hasil uji normalitas dan homogenitas menggunakan analisis non parametrik menghasilkan data pretes berasal dari kelas yang homogen sehingga penelitian dapat dilakukan, dan dari pengujian beda rata-rata menghasilkan kesimpulan bahwa rata-rata kelas pretes dari kelas eksperimen dan kelas kontrol tidak berbeda.

Analisis inferensi untuk data posttest masih menggunakan analisis non parametrik karena data tidak normal menghasilkan kesimpulan bahwa kelas eksperimen dan kelas kontrol pada nilai postes mempunyai rata-rata yang berbeda. Sebenarnya sudah cukup jelas ketika kita melihat hasil rata-rata dari analisis deskriptif dari kelas kontrol dan eksperimen pada pretes dan postes. Analisis inferensi ini untuk menguatkan argumen dan menyimpulkan bahwa dari hasil pretes menyimpulkan yang pertama kedua kelas berasal dari populasi yang sama, sehingga dapat dilakukan penelitian dan yang kedua yaitu hasil postes dari kedua kelas mempunyai rata-rata yang berbeda sehingga dimungkinkan adanya pengaruh penggunaan Indoblockly pada kelas eksperimen untuk itu langkah selanjutnya yaitu mencari bukti apakah ada korelasi dan mencari besaran korelasinya.

4.1.3 Analisis Data Korelasi Pretes Postes Kelas Eksperimen

Sebelum menghitung koefisien korelasi tentu harus dilakukan uji normalitas terhadap data pretes postes kelas eksperimen terlebih dahulu. Hasil dari uji normalitas akan menentukan cara kita untuk menghitung koefisien korelasi.

Jika data pretes postes kelas eksperimen berdistribusi normal maka akan digunakan rumus *Pearson* dan jika data terbukti tidak normal maka akan digunakan rumus *Spearman*.

Untuk menguji normalitas data pretes postes kelas eksperimen, digunakan uji statistik *Shapiro-Wilk*. Perumusan hipotesis pengujian normalitas data pretes postes sebagai berikut:

H₀ : Data sampel berasal dari populasi yang berdistribusi normal.

H₁ : Data sampel berasal dari populasi yang tidak berdistribusi normal.

Dengan menggunakan taraf signifikansi (α) sebesar 0,05 kriteria .

pengujiannya adalah sebagai berikut:

1. Jika nilai *p-value* lebih dari atau sama dengan 0,05 maka H₀ diterima.
2. Jika nilai *p-value* kurang dari 0,05 maka H₀ ditolak.

Adapun hasil dari analisis uji normalitas skor pretes postes kelas eksperimen dengan uji *Shapiro-Wilk* disajikan dalam Tabel 4.10.

Tabel 4.2 Hasil Uji Normalitas Data Pretes Postes kelas eksperimen

Uji	Shapiro-Wilk	
	W	<i>p-value</i>
Pretes	0,83	0,00038
Postes	0,90	0,0146

Berdasarkan Tabel 4.10 dapat dilihat bahwa *p-value* pretes dan postes keduanya $< 0,05$ sehingga disimpulkan bahwa data tidak berdistribusi normal. Dari hasil uji normalitas ternyata terbukti data berasal dari distribusi tidak normal sehingga digunakan rumus *Spearman* untuk menghitung koefisien korelasi.

Sebelum dilakukan analisis korelasi menggunakan *Spearman* dilakukan uji hipotesis korelasi terlebih dahulu. Dengan uji hipotesis korelasi kita dapat mengetahui apakah memang ada korelasi antara hasil belajar dengan penggunaan IndoBlockly pada kelas eksperimen. Dikarenakan hasil uji normalitas menunjukkan bahwa data tidak normal maka untuk menguji hipotesis korelasi digunakan uji

non parametrik dua sampel berhubungan yaitu menggunakan *Wilcoxon*. Perumusan hipotesis pengujian korelasi data pretes sebagai berikut:

H0 : Data pretes dan postes tidak saling berhubungan (berkorelasi).

H1 : Data pretes dan postes saling berhubungan (berkorelasi).

pengujiannya adalah sebagai berikut:

1. Jika nilai *p-value* lebih dari atau sama dengan 0,05 maka H0 diterima.
2. Jika nilai *p-value* kurang dari 0,05 maka H0 ditolak.

Adapun hasil dari analisis uji hipotesis korelasi pretes dan postes kelas eksperimen dengan uji *Wilcoxon* disajikan dalam Tabel 4.10.

Tabel 4.3 Hasil Uji Korelasi Wilcoxon Data Pretes Postes kelas eksperimen

Wilcoxon	
<i>p-value</i>	0,000

Berdasarkan Tabel 4.11 dapat dilihat bahwa *p-value* < 0,05 sehingga disimpulkan bahwa ada korelasi data pretes dan postes kelas eksperimen. Setelah dilakukan uji normalitas dan uji korelasi ternyata diketahui bahwa data tidak normal dan juga ada korelasi maka langkah selanjutnya yaitu menghitung besaran korelasi yaitu dengan *Sprearman*.

Adapun hasil dari perhitungan koefisien korelasi menggunakan *Spearman* dapat dilihat pada Tabel 4.12.

Tabel 4.4 Koefisien Korelasi antara Pretes dan Postes Kelas Eksperimen

	Spearman	
	Pretes	Postes
Pretes	1.0000000	0.4298874
Postes	0.4298874	1.0000000

Berdasarkan hasil output R yang dapat dilihat pada table di atas, diperoleh koefisien korelasi sebesar 0,43 yang berarti bahwa hubungan antara pretes dan postes yang menunjukkan pengalaman belajar menggunakan IndoBlockly mempunyai hubungan yang positif. Berdasarkan Tabel 3.6 yaitu

tabel kriteria korelasi angka 0,43 menunjukkan bahwa ada korelasi yang positif dengan kriteria sedang.

Untuk melakukan uji signifikansi koefisien dapat dilihat dari nilai signifikasinya.

Hipotesis :

H0: Kontribusi variabel independen (X) dan variabel dependen (Y) tidak signifikan (X dan Y independen).

H1: Kontribusi variabel independen (X) dan variabel dependen (Y) signifikan (X dan Y dependen) .

Dengan menggunakan taraf signifikansi 5% maka kriteria pengujiannya adalah ” Jika Nilai $p\text{-value} < \alpha = 0,05$ maka H0 ditolak”. Adapun hasil dari perhitungan uji signifikansi dapat dilihat pada Tabel 4.13.

Tabel 4.5 Uji Signifikansi Koefisien Korelasi antara Pretes dan Postes Kelas Eksperimen

Pretes Postes data eksperimen	Spearman	
	$p\text{-value}$	r
	0.02242	0.4298874

Berdasarkan tabel 4.13 diperoleh nilai signifikansi sebesar $0,02242 < 0,05$, maka H0 ditolak sehingga dapat disimpulkan bahwa Kontribusi variabel independen (X) terhadap variabel dependen (Y) signifikan. Jadi dapat kita simpulkan bahwa ada korelasi positif pembelajaran menggunakan IndoBlockly sebesar 0,43 adalah signifikan dengan kriteria korelasi sedang.

Langkah selanjutnya menghitung besaran hubungan antara pretes dan postes yang menunjukkan pengalaman belajar menggunakan IndoBlockly yaitu dapat dilihat dari koefisien determinasi atau r^2 . Berdasarkan Tabel 4.12 dan

Tabel 4.13 diketahui koefisien korelasi sebesar 0,43, dengan demikian koefisien determinasi atau r^2 diperoleh sebesar 0,1849 atau 18,49 %. Hal ini berarti IndoBlockly memberikan pengaruh sebesar 18,49% terhadap peningkatan hasil belajar mahasiswa pada mata kuliah pemrograman terstruktur.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Uji data pretes dari kelas kontrol dan kelas eksperimen menyatakan bahwa kedua kelas berasal dari populasi yang sama, setelah dilakukan percobaan yaitu dengan memberikan perlakuan yang berbeda terhadap kelas eksperimen menghasilkan data postes yang setelah dianalisis ternyata menunjukkan bahwa data postes kelas kontrol dan kelas eksperimen berbeda. Kelas eksperimen mempunyai skor indeks gain berkriteria sedang yaitu bernilai 0,63 dan koefisien korelasi 0,43. Angka koefisien korelasi 0,43 menunjukkan adanya korelasi positif berkriteria sedang antara penggunaan IndoBlockly terhadap hasil skor postes kelas eksperimen. Hasil uji signifikansi koefisien korelasi menunjukkan bahwa kontribusi variabel independen (X) yaitu penggunaan IndoBlockly terhadap variabel dependen (Y) yaitu hasil belajar mahasiswa adalah signifikan. Besarnya pengaruh penggunaan IndoBlockly terhadap peningkatan hasil belajar mahasiswa pada mata kuliah pemrograman terstruktur sebesar 18,49 %.

5.2 Saran

1. Asisten praktikum pemrograman terstruktur dapat menggunakan IndoBlockly dalam proses kegiatan mengajar karena memang pembelajaran menggunakan IndoBlockly terbukti lebih baik dibandingkan dengan menggunakan model pembelajaran yang lama.
2. Bagi pengembang IndoBlockly hasil penelitian ini adalah sebuah tantangan untuk dapat lebih berinovasi supaya IndoBlockly dapat memberikan pengaruh positif yang lebih besar lagi.
3. Peneliti lain diharapkan dapat melakukan penelitian dengan lingkup yang lebih besar dan sampel yang random sehingga hasil penelitian akan berlaku umum/general.

DAFTAR PUSTAKA

- Anggoro, T .2007. *Metode Penelitian*. Jakarta: Universitas Terbuka.
- Arikunto, Suharsimi .2006. *Prosedur Penelitian Suatu Pendekatan Praktik*. Jakarta:Rineka Cipta.
- Arikunto, Suharsimi. 2007. *Manajemen Penelitian*. Jakarta: Rineka Cipta.
- Arsyad, Azhar. 2004. *Media Pembelajaran*. Jakarta : PT. Raja Grafindo Persada.
- Begel, A,. 1996. *LogoBlocks: A Graphical Programming Language for Interacting with the World*.MIT Media Laboratory
- Bloom B. S.1956. *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York: David McKay Co Inc.
- Boshernitsan, M., Downes, M.2004. *Visual Programming Languages: A Survey*.Computer Science Division (EECS) University of California Berkeley.
- Dahar, R.W. 1989. *Teori-teori Belajar*. Bandung: Erlangga.
- Emzir.2008.*Metodologi Penelitian Pendidikan*. Jakarta: Rajarafindo Persada.
- Ena,O.T.,2007.*Membuat Media Pembelajaran Interaktif dengan Piranti Lunak Presentasi*.Yogyakarta: ILCIC, Universitas Sanata Dharma.
- Erman Suherman dan Yaya Sukjaya K.1990.*Evaluasi Pendidikan Matematika*.Bandung: Wijayakusumah.
- Esteves, M. and Mendes, A., ” *A Simulation Tool to Help Learningof Object Oriented Programming Basics*”. In Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference, Savannah,Georgia, USA, October 2004, 20-23.
- Gay, L.R. dan Diehl, P.L.1992. *Research Methods for Business and Management*. New York :MacMillan Publishing Company.
- Gomes, A. and Carmo, L. and Bigotte, E. and Mendes, A., "*Mathematics and programming problem solving*", 3rd E-Learning Conference – Computer Science Education, Coimbra, September 2006.
- Hundhausen, J. Brown, “*An experimental study of the impact of visual semantic feedback on novice programming*”, Journal of Visual Language and Computing, Vol. 18, 2007, 537-559.

- Jihad dan Haris. 2009. *Evaluasi Pembelajaran*. Yogyakarta: Multi Pressindo
- Kelleher, C. & Pausch, R.,” *Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers*”. ACM Computing Surveys, 37(2), 88-137.
- Lahtinen, E., Mutka, K. A., and Jarvinen, H. M., “*A Study of the difficulties of novice programmers*”, In Proceedings of the 10th annual SIGSCE conference on Innovation and technology in computer science education (ITICSE 2005), Monte da Caparica, Portugal, June 27-29, 2005, ACM Press, New York, NY, pp. 14-18.
- Lethbridge, C.; Diaz-Herrera, J.; LeBlanc, Jr.; Thompson, B., “*Improving software practice through education: Challenges and future trends*”, Future of Software Engineering, (FOSE apos;07), May 2007 Page(s):12 – 28.
- Mafrur, R., “*(IndoBlockly) Visual Programming Editor for Indonesia*”, In Proceedings of Seminar Nasional Ilmu Komputer Universitas Diponegoro (SNIK UNDIP 2012), Semarang, September 15, 2012, Graha Ilmu, Yogyakarta, 155-160.
- Maloney, J., Peppler, K., Kafai, Y., Resnick, M., and Rusk, N., “*Programming by Choice: Urban Youth Learning Programming with Scratch*”. Proceedings of the 39th SIGCSE technical symposium on Computer science education, March 12-15, 2008, Portland, OR, USA .
- Miliszewska, I., Tan, G., “*Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming*”, Journal of Issues in Informing Science & Information Technology, Vol. 4, 2007, 277-289.
- Mustika Danang. 2009. *Matematika Dasar untuk Perguruan Tinggi*. Bandung : Rekayasa Sains.
- Navarro-Prieto, R., Jose J.Can., “*Are visual programming languages better? The role of imagery in program comprehension*”. Departamento de Psicologn H Experimental, Facultad de Psicologn H a, Universidad de Granada.
- Pennington, N. 1987. *Stimulus structures and mental representation in expert comprehension of computer programs*. Cognitive Psychology, 19, 295-341.
- Robins, A, J. Rountree, and N. Rountree. 2003. *Learning and teaching programming: A review and discussion*. Computer Science Education, 13(2):137–172.

- Roscoe, J T. 1992. *Fundamental Research Statistics for the Bahavior al Sciences*. Second Edition Holt. New York :Rinehart and Winston.
- Sugiyono. 2003. *Metode Penelitian Administrasi*. Bandung : CV Alfabeta.
- Sugiyono. 2007. *Statistik untuk Penelitian*. Bandung: Alfabeta.
- Winkel, W.S. 1983 .*Psikologi Pendidikan dan Evaluasi Belajar*. Jakarta: Gramedia
- Winslow, L.E. 1996. *Programming pedagogy – A psychological overview*. SIGCSE Bulletin, 28,17–22.
- Yusniati. 2009. *Pengaruh Model Penemuan Terbimbing Berbasis Konstektual untuk meningkatkan Kemampuan penalaran Matematis Siswa SMP*. Skripsi: Tidak diterbitkan.