

**INTISARI**

**ABSTRACT**



# **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 dilapangan 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 matakuliah 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 matakuliah 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.

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 yang signifikan 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. Aspek yang peneliti tekankan adalah aspek pemahaman terhadap algoritma.
2. Sampel yang diambil adalah mahasiswa semester I Teknik Informatika UIN Sunan Kalijaga tahun 2012/2013 dengan teknik *purposive sampling*.
3. Peneliti tidak melakukan proses karantina terhadap variable kontrol maupun variable percobaan sehingga bisa jadi ada berbagai faktor eksternal yang bisa mempengaruhi kedua variable tersebut.
4. Peneliti tidak menghitung besaran factor external yang peneliti sebutkan pada point ke 3.

### **1.4 Tujuan Masalah**

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 Masalah**

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.

(Robins dkk, 2003) Melakukan sebuah study literatur dari persepektif seorang pengajar yang berkaitan dengan psikologi dan pendidikan pemrograman. Tujuan penelitian adalah untuk mengidentifikasi permasalahan-permasalahan dalam proses belajar programer pemula. Ada beberapa hal yang membuat kami tertarik seperti yang tercantum dalam studi literaturnya 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.

Selanjutnya adalah Penelitian dari Lahtinen dan kawan-kawannya. (Lahtinen dkk, 2005) Melakukan 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.

Yang ketiga yaitu Penelitian yang dilakukan oleh Kelleher dan Pausch. (Kelleher dan Pausch, 2003 ) 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, sintaks 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.

Dari penelitian Kelleher dan Pausch ini respon dari penulis sesuai dengan kondisi penelitian 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 VPL(Visual Programming Languages).

Penelitian yang relevan ke empat yaitu paper dari scratch . (Maloney dkk, 2008) Paper 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 penlulis 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 diikarenakan 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.

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 sudah 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, 2002:12-14), 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 sehari-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, 2002:25-27), 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 (Bloom, 1956) yang secara garis besar membaginya menjadi tiga ranah, yakni ranah kognitif, ranah afektif, dan ranah psikomotorik.

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 pengelompokan 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 menggunkannya. Pengetahuan atau ingatan adalah merupakan proses berfikir yang paling rendah. Salah satu contoh hasil belajar kognitif pada jenjang pengetahuan adalah dapat menghafal jenis-jenis variable 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 menguraikanya 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 (*synthesis*)

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*)**

(Robins dkk, 2003) 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 bahawa dibutuhkan sekitar 10 tahun untuk mengubah programer pemula menjadi programer ahli.

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 programer. Skill/keahlian pemahaman ini akan digunakan programer untuk melakukan tugas-tugas pemrograman seperti *debugging*, *editing* dan, *code reuse*. (Navarro-Prieto, 2000).

Pennington (1987a). Memperkirakan bahwa lebih dari 50% dari semua waktu programer 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 programer, kemajuan



pendidikan, kemajian teknologi desain dan lingkungan pemrograman (programming environments).

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 bagi 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 meyelesaikan masalah dan mebuat algoritma biasanya juga terkendala dalam masalah waktu penyelesaian.

Penyebab lain dari kegagalan belajar pemrograman adalah mahasiswa sudah mengatakan bahawa dia tidak menyukai pemrogramman 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 kemajuannya teknologi dan perkembangan software yang begitu pesat beberapa peneliti mengusulkan design dan software pemrograman berbasis visual dan memang kenyataannya 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*)**

(Boshernitsan dan Downes, 2004) Dari 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 diinterpretasikan dalam bentuk tekstual. Dari berbagai alasan diatas itu membuktikan bahawa penelitian mengenai VPLs itu menarik dan penting.

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

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

##### **2.2.5.1.1 Kelebihan VPLs**

(Begel, 1996) 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).

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**

(Begel, 1996) Di sisi lain, ada juga kerugian. 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

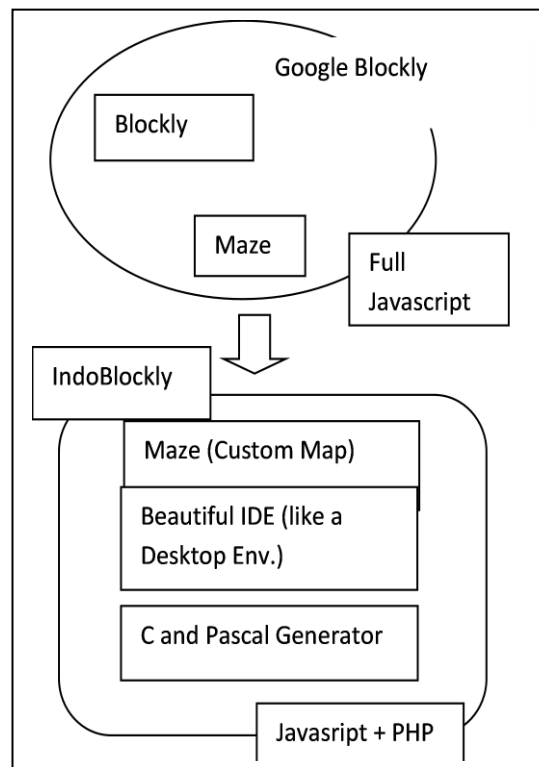
program. Akan tetapi Sebuah bahasa grafis/visual yang terlalu sederhana dapat mengurangi kreatifitas seorang anak untuk membuat program/sesuatu yang menyenangkan. Ketika anak bertambah usia dan lebih cerdas, ia dapat bermigrasi ke bahasa tekstual dan menggunakan kompleksitas yang lebih besar.

### **2.2.6 IndoBlockly**

(Mafrur, 2012) 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 programmer 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.

#### **2.2.6.1 Konsep IndoBlockly**

(Mafrur, 2012) Konsep awalnya ialah kami ingin membangun sebuah software yang bisa membantu anak-anak ataupun siapa saja yang ingin belajar pemrograman menjadi lebih mudah, nyaman, dan senang dalam belajar pemrograman, secara grafis bisa dilihat pada Gambar 2.2., dengan penjelasan sebagai berikut:



**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 Blockly adalah blok-blok puzzle dari *Google Blockly*.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Penelitian**

Menurut Arikunto (2007), “Metode penelitian adalah cara yang digunakan oleh peneliti dalam mengumpulkan data penelitian”. Metode penelitian yang digunakan dalam penelitian ini adalah metode eksperimen. Menurut Arikunto (2006) mengatakan, ”Metode eksperimen yaitu metode penelitian yang dipergunakan untuk meneliti suatu objek dengan melakukan suatu percobaan secara nyata di lapangan”. Penelitian 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, dan sebagainya.



### **3.2.1 Populasi**

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

### **3.2.2 Sampel**

Menurut Anggoro (2007), “Sampel adalah sebagian anggota populasi yang memberikan keterangan atau data yang diperlukan dalam suatu penelitian”. Sampel adalah sebagian atau wakil populasi yang diteliti. 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*)

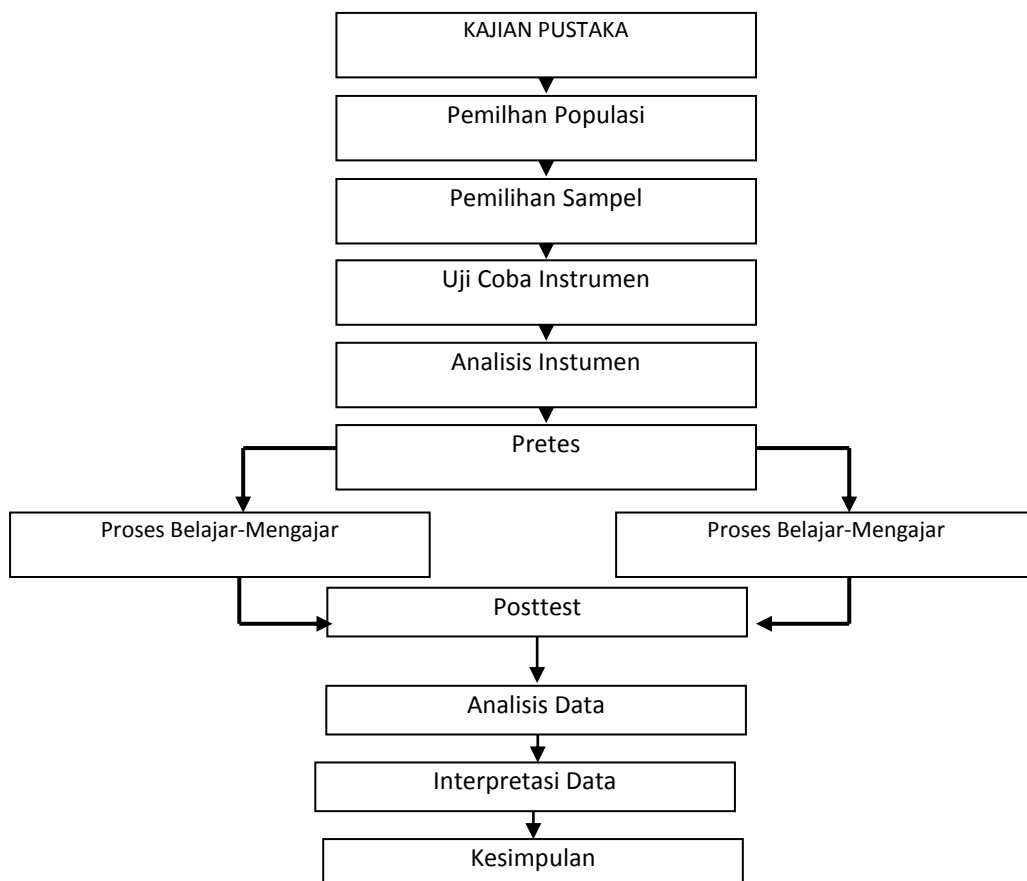
B : Kelompok kontrol (kelompok yang menggunakan model konvensional)

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

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

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 dinyatakan dalam diagram alur pada 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.

### **3.5.3 Penyelesaian Penelitian**

Setelah penelitian selesai dilakukan langkah selanjutnya yaitu 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. Adapun pedoman dalam menganalisa soal tes uji coba adalah sebagai berikut:

### 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. Menurut Suherman dan Sukjaya (1990) Validitas butir soal essay (uraian) dihitung dengan menggunakan rumus koefisien korelasi *Product Moment* memakai angka kasar, yaitu:

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

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 (1990) untuk menentukan reliabilitas soal berbentuk essay (uraian) digunakan rumus *Alpha*, yaitu:

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

Keterangan:

$r_{11}$  = koefisien reliabilitas instrumen

$n$  = banyaknya butir soal

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

$S_i^2$  = 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: Suheman dan Sukjaya (1990: 177)

### 3.6.1.3 Menentukan Daya Pembeda Soal

Pengertian Daya Pembeda dari sebuah butir soal menurut Suherman dan Sukjaya (1990) “Seberapa jauh kemampuan butir soal tersebut mampu membedakan antara testi yang mengetahui jawabannya dengan benar dengan testi yang tidak dapat menjawab soal tersebut”. Daya pembeda dihitung dengan menggunakan rumus DP untuk tes uraian menurut Jihad dan Haris (2009) 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 ( <i>DP</i> )	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: 202)

### 3.6.1.4 Menentukan Tingkat Kesukaran Soal

Untuk menghitung tingkat kesukaran soal yang berbentuk uraian menurut

Jihad dan Haris (2009) 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: 213)

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 penulis memutuskan untuk menggunakan tujuh soal yaitu soal nomor 1,2,3,4,5,6, dan 7 dalam pretes dan postes.

Adapun hasil perhitungan instrumen disajikan pada Tabel 3.6 sebagai berikut:

**Tabel 3.6**  
**Hasil Analisis Instrumen Tes Uji Coba**

No Soal	Validitas		Reliabilitas		Daya Pembeda		Indeks Kesukaran	
	Indeks	Interpretasi	Indeks	Interpretasi	Indeks	Interpretasi	Indeks	Interpretasi
1	0,7126652	Tinggi	0,597349	Sedang	0,43	Baik	0,44	Sedang
2	0,6850126	Tinggi			0,24	Sedang	0,27	Sedang
3	0,9131264	Sangat Tinggi			0,33	Sedang	0,46	Sedang
4	0,6340893	Tinggi			0,35	Sedang	0,49	Sedang
5	0,9099195	Sangat Tinggi			0,47	Baik	0,49	Sedang
6	0,9387633	Sangat Tinggi			0,39	Sedang	0,51	Sedang
7	0,6942779	Tinggi			0,28	Sedang	0,35	Sedang
8	0,5049758	Sedang			0,09	Jelek	0,15	Sulit
9	0,5476324	Sedang			0,16	Jelek	0,23	Sulit
10	0,4252906	Sedang			0,19	Jelek	0,19	Sulit

### 3.7 Teknik Pengumpulan Data

Pengumpulan data dilakukan pada awal penelitian yaitu menggunakan pretes dan akhir penelitian yaitu menggunakan postes. Secara garis besar teknik pengumpulan data dalam penelitian ini dapat disajikan dalam tabel 3.7 sebagai berikut:

**Tabel 3.7**  
**Teknik Pengumpulan Data**

No	Jenis data	Teknik Pengumpulan
1	Data kuantitatif dapat dilihat dari hasil belajar mahasiswa.	Tes (pretes dan postes)

### 3.8 Teknik Pengolahan Data

Data yang diperoleh dalam penelitian ini berasal dari tes (pretes dan postes) yang berupa soal uraian. Data-data yang diperoleh adalah sebagai berikut:

#### 3.8.1 Pengolahan Data Kuantitatif

Ada dua macam data yang bisanya digunakan dalam penelitian yaitu data kuantitatif dan kualitatif. Penelitian ini hanya menggunakan dan menganalisis hasil perolehan skor pretes dan postes yang berupa data kuantitatif. Hasil dari analisis data penelitian ini adalah sebagai berikut:

##### 3.8.1.1 Analisis Data Tes

Analisis data tes dilakukan untuk menjawab rumusan masalah apakah ada pengaruh pembelajaran menggunakan IndoBlockly dibandingkan dengan pembelajaran konvensional pada praktikum pemrograman terstruktur. Selain dari hasil pretes dan postes, data kuantitatif juga diperoleh dari gain kedua kelas. Gain yang dimaksud dalam penelitian ini adalah gain

yang ternormalisasi atau *Normalized Gain (NG)*. *Normalized Gain* adalah proporsi gain actual (postes-pretes) dengan gain maksimal yang dicapai (Mustika, 2009). Rumus yang digunakan untuk menjelaskan gain dibuat oleh Hake (Mustika, 2009:), yaitu:

$$NG = \frac{POS - PRE}{IDEAL - PRE}$$

Keterangan:

*GN* = Gain yang ternormalisasi

*POS* = Skor postes mahasiswa

*PRE* = Skor pretes mahasiswa

*IDEAL* = Skor ideal / Nilai maksimum

**Tabel 3.5 Kategori Gain yang dinormalisasi**

No	Normalized Gain	Kriteria
1	$NG \geq 0,70$	Tinggi
2	$0,30 < NG \leq 0,70$	Sedang
3	$NG < 0,30$	Rendah

Langkah-langkah yang dilakukan dalam teknik analisis data tes, baik pretes, postes, maupun indeks gain adalah sebagai berikut:

#### **3.8.1.1.1 Analisis Deskriptif**

Analisis deskriptif dilakukan untuk memperoleh gambaran umum mengenai data pretes dan data postes dari kelas eksperimen dan kelas kontrol yang diperoleh berupa skor rata-rata (mean) dan standar deviasi.

#### **3.8.1.1.2 Analisis Inferensi**

Analisis inferensi dilakukan untuk memperoleh kesimpulan ada atau tidaknya pengaruh pembelajaran menggunakan IndoBlockly pada kelas eksperimen dan dibandingkan dengan kelas kontrol yang menggunakan model pembelajaran konvensional. Analisis ini pada intinya merupakan uji perbedaan dua rata-rata, baik uji dua pihak maupun satu pihak. Sebelum melakukan uji perbedaan dua rata-rata, terlebih dahulu dilakukan uji normalitas dan homogenitas. Uji normalitas dan uji homogenitas dipandang perlu dilakukan karena dengan dilakukannya uji normalitas dan homogenitas, langkah-langkah penelitian dapat dipertanggungjawabkan dan kesimpulan yang dibuat berdasarkan teori dapat berlaku (Yusniati, 2009). Pengolahan data penganalisisan data hasil penelitian dilakukan dengan bantuan *software R* dan *Microsoft Excel 2010*. Adapun langkah-langkah analisis inferensi adalah sebagai berikut:

##### **3.8.1.1.2.1 Uji Normalitas**

Uji ini dilakukan untuk mengetahui data dari masing-masing kelompok sampel berdistribusi normal atau tidak. Data-data yang diuji adalah data pretes kelas kontrol, pretes kelas eksperimen, postes kelas kontrol, postes kelas eksperimen, gain kelas kontrol dan gain kelas eksperimen. Dalam uji normalitas ini digunakan uji *Shapiro –Wilk*. Jika data berasal dari distribusi yang normal, maka analisa data dilanjutkan dengan uji homogenitas varians untuk menentukan uji parametrik yang sesuai. Namun, jika data berasal dari populasi yang tidak berdistribusi normal, maka tidak dilakukan uji homogenitas varians tetapi langsung dilakukan uji kesamaan dua rata-rata (uji non-parametrik) yaitu dengan menggunakan *Mann Whitney U*.

#### **3.8.1.1.2.2 Uji Homogenitas Varians**

Uji homogenitas varians dilakukan jika data yang diolah berdistribusi normal. Uji homogenitas ini dilakukan untuk mengetahui apakah variansi populasi data yang diuji memiliki variansi yang homogen atau tidak. Untuk menguji homogenitas varians digunakan uji *Lavene's Test* dengan mengambil taraf kepercayaan 95% (taraf signifikansi 5%). Jika data yang telah dianalisis bersifat normal dan homogen, maka data tersebut dilakukan uji perbedaan dua rata-rata.

#### **3.8.1.1.2.3 Uji Perbedaan Dua Rata-rata**

Uji perbedaan dua rata-rata yang dilakukan yaitu untuk menguji apakah terdapat perbedaan rata-rata (*mean*) pretes dan postes antara kelas eksperimen dan kelas kontrol. Uji perbedaan dua rata-rata terhadap skor pretes dilakukan dengan menggunakan uji dua pihak dan uji perbedaan dua rata-rata terhadap skor postes dilakukan dengan menggunakan uji satu pihak. Jika data telah berdistribusi normal dan homogen, maka dilakukan pengujian perbedaan dua rata-rata dengan menggunakan uji-t. Sedangkan untuk data yang berdistribusi normal tetapi tidak homogen, maka dilakukan pengujian perbedaan dua rata-rata dengan menggunakan uji-t'. Adapun data yang tidak berdistribusi normal maka pengujiannya menggunakan uji non-parametrik yaitu uji *Mann Whitney U*. Jika rata-rata pretes kelas eksperimen dan kelas kontrol secara statistik tidak sama, maka untuk menguji peningkatan kemampuan penalaran adaptif siswa diambil dari *indeks gain*.

## **BAB IV**

### **HASIL PENELITIAN DAN PEMBAHASAN**

Pada bab ini akan dibahas mengenai analisis data dari hasil pengolahan data-data yang diperoleh dari hasil penelitian. Hasil analisis data yang diperoleh merupakan gambaran keseluruhan hasil dari penelitian yang telah dilaksanakan. Data yang diperoleh dari penelitian ini yaitu data kuantitatif. Data kuantitatif merupakan data yang diperoleh dari hasil tes yang berupa pretes dan postes. Dari hasil pretes dan postes diperoleh data kuantitatif lainnya yaitu data gain. Pengolahan data dilakukan menggunakan software *Statistic R Tool For Windows* dan *Microsoft Office Excel 2010*.

#### **4.1 Hasil Penelitian dan Pembahasan**

Soal pretes dan postes yang tadinya berjumlah sepuluh butir soal kemudian dilakukan uji validitas, reabilitas, daya beda, dan tingkat kesukaran menghasilkan tujuh soal yang valid. Kemudian tujuh soal tadi dijadikan sebagai soal pretes dan postes. Ujian pretes dilakukan pertemuan pertama kegiatan belajar mengajar sedangkan soal postes diujikan setelah dilakukan lima kali pertemuan proses pembelajaran. Hasil skor pretes dan postes sudah dianalisis dengan hasil sebagai berikut:

##### **4.1.1 Analisis Deskriptif Data Hasil Pretes dan Postes**

Pada bab sebelumnya telah dijelaskan bahwa analisis deskriptif dilakukan untuk memperoleh gambaran umum mengenai data pretes dan data postes dari kelas eksperimen dan kelas kontrol yang diperoleh berupa skor rata-rata (mean) dan standar deviasi. Untuk



mengetahui kemampuan awal mahasiswa sebelum diberi perlakuan pembelajaran dengan menggunakan IndoBlockly, maka pada kelas eksperimen dan kelas kontrol diberikan soal pretes untuk mengetahui kemampuan mahasiswa baik dari kelas eksperimen maupun kelas kontrol. Sedangkan untuk mengetahui kemampuan akhir mahasiswa serta untuk melihat peningkatan kemampuan mahasiswa terhadap pemahaman algoritma, maka pada kelas eksperimen dan kelas kontrol diberikan soal postes. Selanjutnya, untuk mengetahui kualitas peningkatan kemampuan mahasiswa dilakukan analisis data gain kelas eksperimen dan juga kelas kontrol. Berdasarkan data yang diperoleh, gambaran umum tersebut dapat dilihat pada Tabel 4.1 di bawah ini.

**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. Hasil tersebut memberikan gambaran bahwa terdapat selisih yang cukup besar antara rata-rata kemampuan akhir mahasiswa antara kelas eksperimen dan kelas kontrol. Selain perbedaan rata-rata kemampuan awal dan perbedaan rata-rata kemampuan akhir pada kedua kelas, hasil tersebut juga memberikan gambaran bahwa penyebaran data pada kedua kelas berbeda, di mana skor pretes kelas eksperimen lebih menyebar (bervariasi) dari pada skor pretes kelas kontrol. Akan tetapi, skor postes kelas kontrol lebih menyebar dari pada skor postes kelas eksperimen. Perbedaan penyebaran data tersebut dapat dilihat dengan adanya perbedaan standar deviasi pada kedua kelas dan batas nilai maksimum dan minimum kedua kelas, baik pada skor pretes maupun skor postes. Namun demikian, untuk mengetahui apakah rata-rata kemampuan awal dan akhir mahasiswa kedua kelas berbeda secara signifikan atau tidak, akan dilakukan uji statistik. Untuk menguji hal tersebut, berikut ini dilakukan analisis inferensi terhadap data hasil dari pretes dan postes mahasiswa.

#### **4.1.2 Analisis Inferensi Data Hasil Pretes dan Postes**

Seperti yang telah diuraikan pada Bab III, analisis inferensi dilakukan untuk memperoleh kesimpulan apakah model pembelajaran menggunakan IndoBlockly yang diterapkan pada mahasiswa kelas eksperimen lebih baik daripada kelas kontrol yang menggunakan model pembelajaran biasa/konvensional. Analisis ini pada intinya merupakan uji perbedaan dua rata-rata, baik uji satu pihak maupun dua pihak.

#### 4.1.2.1 Analisis Data Kemampuan Awal Mahasiswa

Analisis data kemampuan awal mahasiswa ini diperoleh dari data pretes. Sesuai dengan tahap-tahap pelaksanaan penelitian, sebelum pembelajaran pada kelas eksperimen dan kelas kontrol berlangsung dilaksanakan tes awal (pretes). Tujuan diberikannya pretes pada masing-masing kelas adalah untuk mengetahui kemampuan awal mahasiswa pada kelas eksperimen dan kelas kontrol sama atau berbeda. Hasil pretes mahasiswa dari kelas kontrol dan kelas eksperimen dapat dilihat pada Lampiran D.1, berikut ini disajikan analisis deskriptif data pretes kelas eksperimen dan kelas kontrol.

**Tabel 4.2**  
**Data Statistik Skor Pretes Kelas Eksperimen dan Kelas Kontrol**

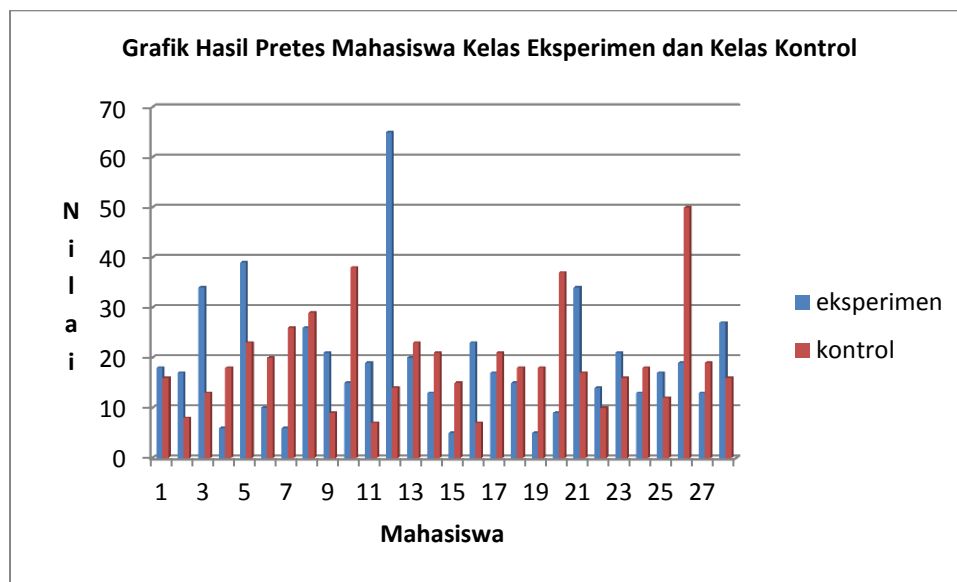
Kelas	Banyak Mhs	Pretes				
		Min	Max	Mean	Sd	variance
Eksperimen	28	5,00	65,00	19,32	12,46	155,25
Kontrol	28	7,00	50,00	19,25	9,78	95,65

Berdasarkan Tabel 4.2 terlihat bahwa rata-rata skor pretes yang diperoleh mahasiswa kelas eksperimen dan mahasiswa kelas kontrol hampir sama. Hal ini menunjukkan bahwa kemampuan awal mahasiswa baik kelas eksperimen maupun kelas kontrol hampir sama secara eksak. Secara grafik hasil pretes kemampuan mahasiswa antara kelas eksperimen dan kelas kontrol dapat dilihat pada Gambar 4.1.

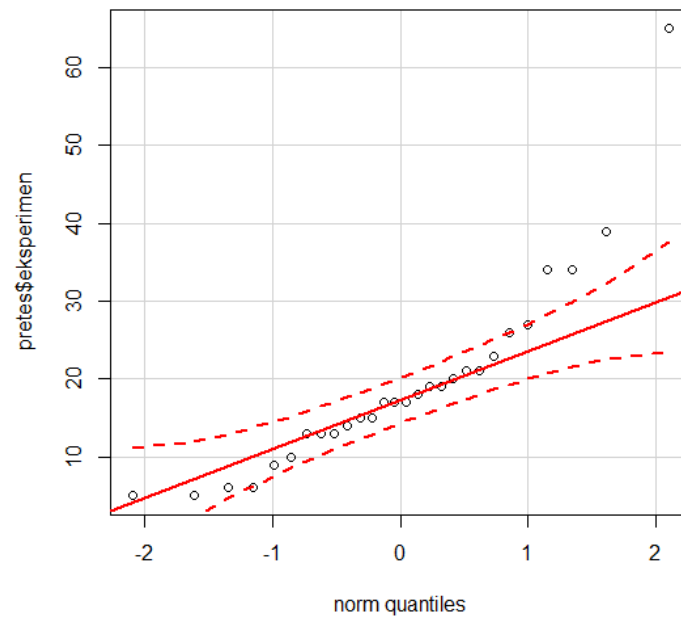
Sekilas melihat dari hasil rata-rata tersebut bisa dikatakan bahwa kelas eksperimen dan kelas kontrol homogen (berasal dari populasi yang sama), tapi jelas ini harus dibuktikan dengan uji statistik terlebih dahulu. Uji statistik yang pertama yaitu uji normalitas, dari hasil dari uji

normalitas kita bisa menentukan langkah-langkah analisis data berikutnya. Jika data terbukti normal jelas kita akan menggunakan analisis data parametrik, tapi jika data ternyata tidak normal maka akan digunakan metode analisis data non parametrik. Pengujian konormalan bisa disajikan menggunakan Q-Q Plot seperti pada Gambar 4.2 dan Gambar 4.3.

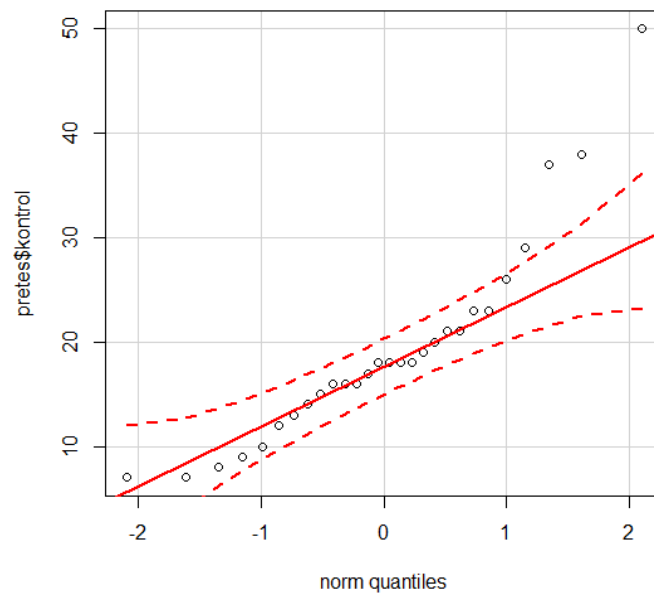
Gambar 4.1 dan Gambar 4.2 terlihat bahwa sebaran data pretes tidak tersebar pada garis lurus, sehingga dapat diduga bahwa data sampel kelas eksperimen dan kelas kontrol berasal dari populasi yang berdistribusi tidak normal. Namun untuk mengetahui apakah prediksi tersebut itu benar, maka selanjutnya dilakukan uji normalitas.



**Gambar 4.1 Hasil Pretes Kelas Eksperimen dan Kelas Kontrol**



**Gambar 4.2 .Q-Q plot pretes kelas eksperimen**



**Gambar 4.3. Q-Q plot pretes kelas kontrol**

#### 4.1.2.1.1 Uji Normalitas Data Pretes

Langkah pertama yang dilakukan adalah dengan menguji normalitas data pretes kedua kelas, untuk mengetahui apakah kedua kelas tersebut berdistribusi normal atau tidak. Untuk menguji normalitas data pretes, digunakan uji statistik *Shapiro-Wilk*. Perumusan hipotesis pengujian normalitas data pretes sebagai berikut:

$H_0$  : Skor pretes sampel berasal dari populasi yang berdistribusi normal.

$H_1$  : Skor pretes sampel berasal dari populasi yang tidak berdistribusi normal.

Dengan menggunakan taraf signifikansi ( $\alpha$ ) sebesar 0,05 kriteria pengujiannya adalah sebagai berikut:

1. Jika nilai *p-value* lebih dari atau sama dengan 0,05 maka  $H_0$  diterima.
2. Jika nilai *p-value* kurang dari 0,05 maka  $H_0$  ditolak.

Adapun hasil dari analisis uji normalitas skor pretes kelas eksperimen dan kelas kontrol dengan uji *Shapiro-Wilk* menggunakan *R statistic tool* disajikan dalam Tabel 4.3 berikut ini.

**Tabel 4.3**  
**Hasil Uji Normalitas Data Pretes**

Kelas	Shapiro-Wilk	
	W	p-value
Eksperimen	0,83	0,0003
Kontrol	0,87	0,003

Dari Tabel 4.3 terlihat bahwa (p-value) uji *Shapiro-Wilk* untuk kelas eksperimen adalah 0,0003 dan kelas kontrol adalah 0,003. Nilai p-value baik kelas eksperimen maupun kelas kontrol kurang dari 0,05 maka  $H_0$  ditolak, artinya data pretes kelas eksperimen dan kelas kontrol tidak berdistribusi normal. Hasil pengujian menggunakan uji *Shapiro-Wilk*

menunjukkan hasil bahwa data hasil dari pretes kedua kelas terbukti tidak normal sehingga pengujian yang dilakukan selanjutnya adalah uji homogenitas. Dikarenakan hasil dari uji normalitas menunjukkan bahwa data berdistribusi tidak normal maka digunakan uji kesamaan dua rata-rata non-parametrik yaitu uji *Mann Whitney*.

#### **4.1.2.1.2 Uji Perbedaan Dua Rata-rata Data Pretes**

Uji kesamaan dua rata-rata dalam penelitian ini menggunakan uji *Mann-Whitney*.

Hipotesis dalam pengujian kesamaan dua rata-rata dirumuskan sebagai berikut:

H<sub>0</sub> : Tidak terdapat perbedaan kemampuan awal mahasiswa antara kelas eksperimen dengan kelas kontrol.

H<sub>1</sub> : Terdapat perbedaan kemampuan awal mahasiswa antara kelas eksperimen dengan kelas kontrol.

Pasangan hipotesis tersebut jika dirumuskan ke dalam bentuk hipotesis statistik adalah sebagai berikut:

$$H_0: \mu_e = \mu_k$$

$$H_1: \mu_e \neq \mu_k$$

Keterangan:

$\mu_e$  = rata-rata nilai pretes kelas eksperimen.

$\mu_k$  = rata-rata nilai pretes kelas kontrol.

Dengan menggunakan taraf signifikansi ( $\alpha$ ) adalah 0,05 maka kriteria pengujiannya sebagai berikut:

1. Jika nilai *p-value* lebih dari atau sama dengan 0,05 maka H<sub>0</sub> diterima.
2. Jika nilai *p-value* kurang dari 0,05 maka H<sub>0</sub> ditolak.

Hasil analisis uji *Mann-Whitney* skor pretes disajikan dalam Tabel 4.4 berikut ini.

**Tabel 4.4**  
**Hasil Uji Mann-Whitney Data Pretes**

Mann-Whitney	Nilai
W	411,5
p-value	0,749

Pada Tabel 4.4 diperoleh bahwa nilai p-value *Mann-Whitney* adalah 0,749. Karena 0,749 lebih dari 0,05 maka berdasarkan kriteria pengujian di atas  $H_0$  diterima. Hal ini menunjukkan bahwa tidak terdapat perbedaan rata-rata skor pretes antara kelas eksperimen dan kelas kontrol. Dengan kata lain, kemampuan awal mahasiswa kelas eksperimen dengan kelas kontrol adalah sama.

#### **4.1.2.2 Analisis Data Kemampuan Akhir Mahasiswa**

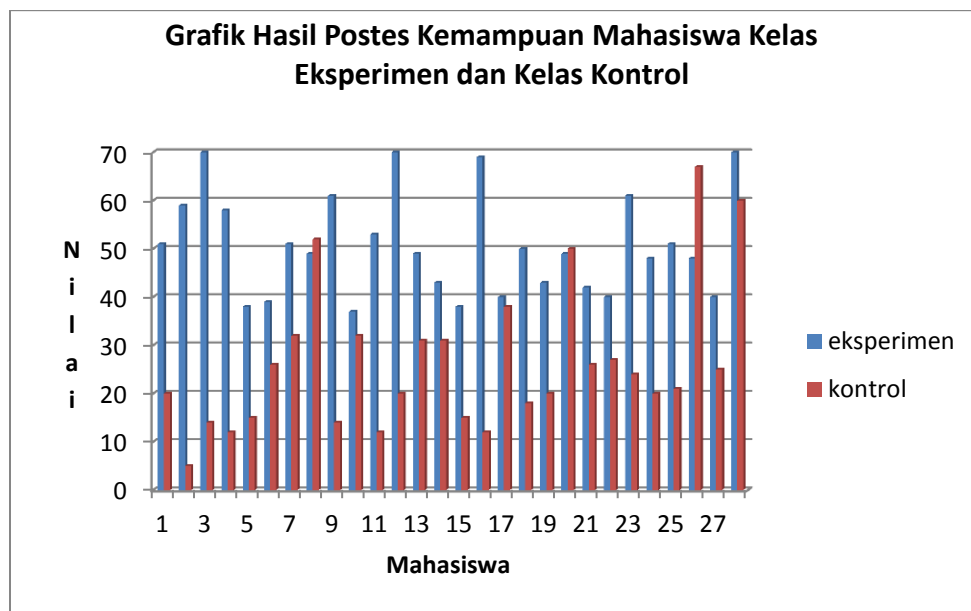
Berdasarkan analisis data pretes diperoleh kesimpulan tidak terdapat perbedaan rata-rata yang signifikan kemampuan awal mahasiswa antara kelas eksperimen dan kelas kontrol, atau dengan kata lain kemampuan awal mahasiswa pada kedua kelas tersebut adalah sama. Maka penelitian ini layak untuk dilanjutkan dikarenakan memang terbukti bahwa kedua kelas yaitu kelas eksperimen dan kelas kontrol tidak terdapat perbedaan rata-rata yang signifikan.. Selanjutnya adalah analisis data kemampuan akhir mahasiswa atau analisis data postes. Hasil uji kemampuan mahasiswa dari kelas eksperimen dan kelas kontrol yaitu analisis data hasil postes dapat dilihat pada Lampiran D.2, berikut ini disajikan analisis deskriptif data postes kelas eksperimen dan kelas kontrol.



**Tabel 4.5**  
**Statistika Deskriptif Skor Postes Kelas Eksperimen dan Kelas Kontrol**

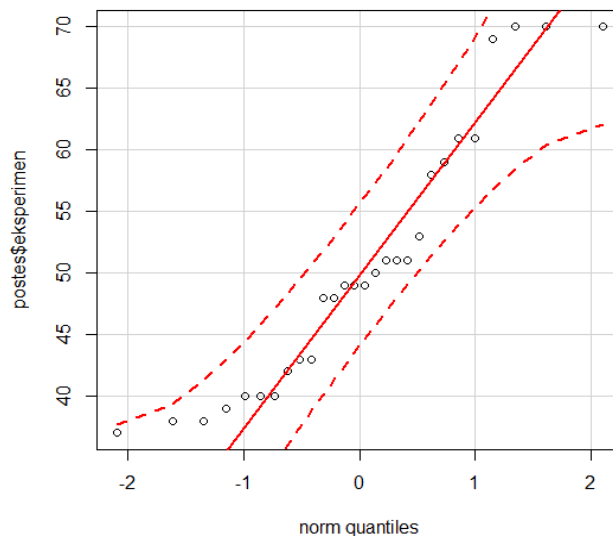
Kelas	Banyak Mhs	Postes				
		Min	Max	Mean	Sd	variance
Eksperimen	28	37,00	70,00	50,61	10,54	111,09
Kontrol	28	5,00	67,00	26,39	15,09	227,71

Berdasarkan Tabel 4.5 memberikan gambaran bahwa rata-rata kemampuan mahasiswa kelas eksperimen berbeda dengan kelas kontrol. Secara grafik hasil postes kemampuan mahasiswa dapat dilihat bahwa terdapat perbedaan rata-rata kemampuan akhir mahasiswa antara kelas eksperimen dan kelas kontrol pada Gambar 4.4 . Berdasarkan Gambar 4.4 terlihat bahwa diagram batang yang berwarna biru yaitu diagram nilai postes kelas eksperimen rata-rata memiliki nilai yang lebih besar dibandingkan dengan diagram batang yang berwarna merah tua yaitu diagram batang nilai postes kelas kontrol.

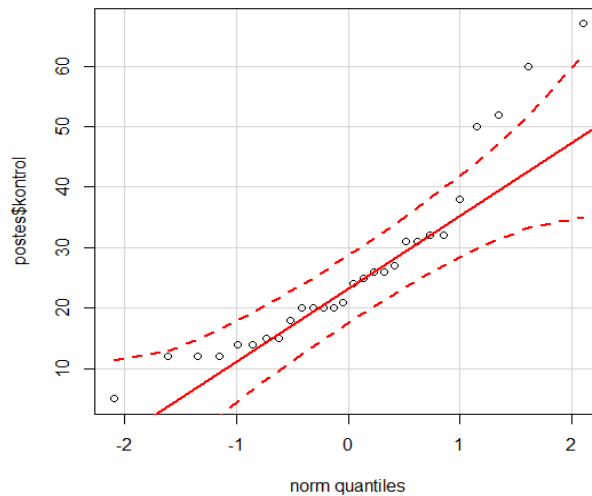


**Gambar 4.4 Hasil Postes Kelas Eksperimen dan Kelas Kontrol**

Langkah berikutnya yaitu uji normalitas data postes, berikut ini kami sajikan Q-Q plot uji kenormalan data postes kelas eksperimen dan kelas kontrol. Gambar 4.5 yaitu Q-Q plot dari postes kelas eksperimen sebaran data postes tersebar mendekati garis lurus, sehingga dapat diduga data berdistribusi normal. Sedangkan sebaran data postes kelas kontrol yang disajikan dalam bentuk Q-Q plot pada Gambar 4.6 terlihat bahwa sebaran data postes tersebar menjauh dari garis lurus, sehingga dapat diduga bahwa data sampel kelas kontrol berasal dari populasi yang tidak berdistribusi normal. Karena ada salah satu kelas yang tidak berdistribusi normal, maka data postes kedua kelas tersebut berasal dari populasi yang tidak berdistribusi normal. Namun untuk mengetahui apakah prediksi tersebut itu benar, maka selanjutnya dilakukan uji statistik dengan langkah-langkah sebagai berikut.



**Gambar 4.5 Uji Normalitas dengan Q-Q plot data postes kelas eksperimen**



**Gambar 4.6 Uji Normalitas dengan Q-Q plot data postes kelas kontrol**

#### **4.1.2.2.1 Uji Normalitas Data Postes**

Sama halnya dengan uji normalitas data pretes, langkah pertama yang dilakukan adalah menguji normalitas data postes kedua kelas untuk mengetahui apakah kedua kelas tersebut berdistribusi normal atau tidak. Untuk menguji normalitas data postes, digunakan uji statistik *Shapiro-Wilk*. Perumusan hipotesis pengujian normalitas data pretes sebagai berikut:

$H_0$  : Data sampel berasal dari populasi yang berdistribusi normal.

$H_1$  : Data sampel berasal dari populasi yang tidak berdistribusi normal. Dengan menggunakan taraf signifikansi ( $\alpha$ ) sebesar 0,05 kriteria .

pengujiannya adalah sebagai berikut:

1. Jika nilai *p-value* lebih dari atau sama dengan 0,05 maka  $H_0$  diterima.
2. Jika nilai *p-value* kurang dari 0,05 maka  $H_0$  ditolak.

Adapun hasil dari analisis uji normalitas skor pretes kelas eksperimen dan kelas kontrol dengan uji *Shapiro-Wilk* disajikan dalam Tabel 4.6 berikut ini.

**Tabel 4.6**  
**Hasil Uji Normalitas Data Postes**

Kelas	Shapiro-Wilk	
	W	p-value
Eksperimen	0,90	0,0146
Kontrol	0,88	0,0038

Dari Tabel 4.6 terlihat bahwa (p-value) uji *Shapiro-Wilk* untuk kelas eksperimen adalah 0,0146 dan kelas kontrol adalah 0,0038. Nilai p-value kelas eksperimen lebih dari 0,05 maka  $H_0$  diterima, artinya data postes kelas eksperimen berasal dari populasi yang berdistribusi normal. Sedangkan nilai p-value untuk kelas kontrol kurang dari 0,05 maka  $H_0$  ditolak, artinya data postes kelas kontrol berasal dari populasi yang tidak berdistribusi normal. Karena ada salah satu sampel yang tidak berdistribusi normal, maka tidak dilakukan uji homogenitas varians. Sehingga pengujian yang dilakukan selanjutnya adalah uji kesamaan dua rata-rata dengan menggunakan uji non-parametrik yaitu uji *Mann-Whitney*.

#### **4.1.2.2.2 Uji Perbedaan Dua Rata-rata Data postes**

Berdasarkan hasil dari uji normalitas yang telah dilakukan, diketahui bahwa data postes berasal dari populasi yang tidak berdistribusi normal, maka selanjutnya dilakukan analisis dengan menggunakan statistika nonparametrik, yaitu uji *Mann-Whitney*. Perumusan hipotesis untuk uji perbedaan dua rata-rata skor postes ini adalah sebagai berikut:

$H_0$  : Kemampuan mahasiswa kelas eksperimen tidak lebih baik daripada kemampuan mahasiswa kelas kontrol.

H1 : Kemampuan mahasiswa kelas eksperimen lebih baik daripada kemampuan mahasiswa kelas kontrol.

Kriteria pengambilan keputusan untuk pengujiannya adalah sebagai berikut:

- a) Jika nilai *p-value* lebih besar atau sama dengan 0,05 maka H0 diterima.
- b) Jika nilai *p-value* lebih kecil dari 0,05 maka H0 ditolak.

Hasil analisis uji *Mann-Whitney* skor postes disajikan dalam Tabel 4.7 berikut ini.

Tabel 4.7  
Hasil Uji Mann Whitney Data Postes

Mann-Whitney	Nilai
W	66,5
p-value	0,000

Dari Tabel 4.7 terlihat bahwa nilai *p-value* adalah 0,000 kurang dari 0,05 maka berdasarkan kriteria pengujian H0 ditolak. Dengan kata lain, kemampuan mahasiswa dalam pemahaman algoritma pada praktikum pemrograman terstruktur yang menggunakan IndoBlockly lebih baik secara signifikan dari pada mahasiswa yang menggunakan pembelajaran konvensional.

#### **4.1.2.3 Analisis Data Kualitas Peningkatan Kemampuan Mahasiswa**

Variabel yang diukur dalam penelitian ini adalah peningkatan kemampuan mahasiswa yang mendapat model pembelajaran menggunakan IndoBlockly. Kemampuan mahasiswa antara kelas eksperimen dengan kelas kontrol setelah pembelajaran sudah diketahui pada analisis postes dengan kesimpulan bahwa kemampuan mahasiswa pada kelas eksperimen lebih baik daripada mahasiswa kelas kontrol. Untuk mengetahui kualitas peningkatan

kemampuan mahasiswa kelas eksperimen dan kelas kontrol, dilakukan uji statistik deskriptif terhadap skor indeks gain tes kemampuan mahasiswa kelas eksperimen dan kelas kontrol sebagai berikut:

**Tabel 4.8**  
**Statistik Deskriptif Skor Indeks Gain Tes Kemampuan Mahasiswa Kelas Eksperimen dan Kelas Kontrol**

Kelas	N	Mean Gain	Kriteria
Eksperimen	28	0,625128218	Sedang
Kontrol	28	0,159511988	Rendah

Dari Tabel 4.8 di atas, terlihat bahwa rata-rata indeks gain tes kemampuan mahasiswa kelas eksperimen adalah 0,63 dan rata-rata indeks gain tes kemampuan mahasiswa untuk kelas kontrol adalah 0,16. Berdasarkan kriteria indeks gain menurut Hake, ini berarti bahwa kualitas peningkatan kemampuan mahasiswa kelas eksperimen tergolong sedang sedangkan kelas kontrol tergolong rendah. Berikut ini adalah daftar persentase kualitas peningkatan kemampuan mahasiswa kelas eksperimen dan kelas kontrol.

**Tabel 4.9**  
**Daftar Persentase Kualitas Peningkatan Kemampuan Mahasiswa Kelas Eksperimen dan Kelas Kontrol**

Kualitas Peningkatan Kemampuan Mahasiswa	Kelas Eksperimen		Kelas Kontrol	
	Jumlah Mhs	Persentase	Jumlah Mhs	Persentase
Tinggi	9	32,14	2	7,14
Sedang	17	60,71	3	10,71
Rendah	2	7,14	23	82,14

Berdasarkan Tabel 4.9 terlihat bahwa untuk kelas eksperimen 60,71% dari jumlah (28) mahasiswa termasuk dalam kategori sedang dan 32,14% lainnya termasuk dalam kategori tinggi. Hal ini menunjukkan bahwa peningkatan kemampuan mahasiswa yang terjadi di

kelas eksperimen termasuk dalam kategori sedang, dimana lebih dari 50% (60,71%) mahasiswa kelas eksperimen mengalami peningkatan kemampuan yang sedang dan (32,14%) mengalami peningkatan yang tinggi cukup banyak jika dibandingkan dengan kelas kontrol yang hanya (7,14%).

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

#### **5.2 Saran**



## DAFTAR PUSTAKA

- A. Robins, J. Rountree, and N. Rountree.(2003).*Learning and teaching programming: A review and discussion*. Computer Science Education, 13(2):137–172.
- 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.

- 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. (2005). *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. 2012. *"(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. (2008). *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.(2001).*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. (1987a). *Stimulus structures and mental representation in expert comprehension of computer programs*. Cognitive Psychology, 19, 295-341.

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