

BAB IV

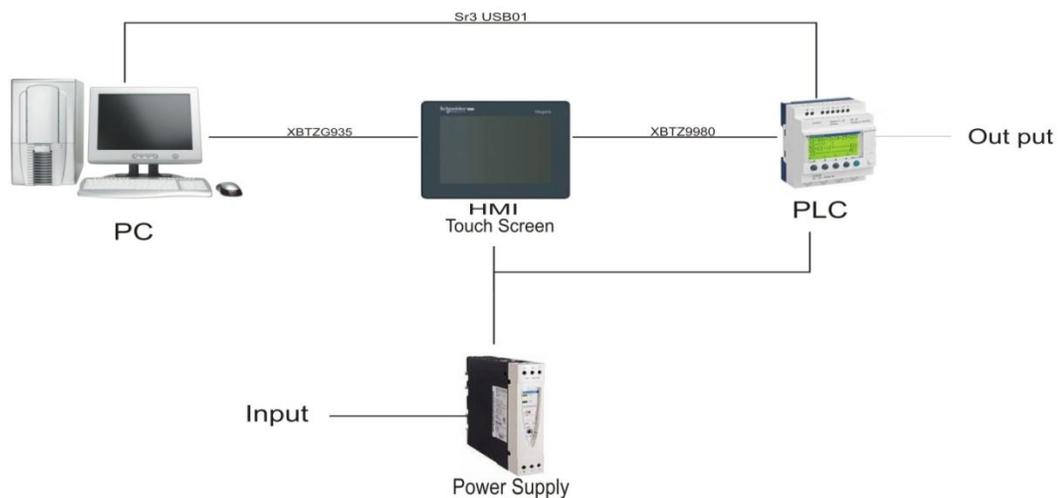
PERANCANGAN DAN PEMBAHASAN

Pada bab ini akan diterangkan secara detail mengenai perancangan trainer simulator HMI berbasis PLC. Simulator ini memiliki beberapa bagian penting yaitu perancangan *hardware* dan pemograman *software*, untuk pemograman *software* menggunakan *logic diagram (Leader diagram)* berbasis PLC, untuk pemogramam itu sendiri terdapat dua bagian, pertama pemograman HMI (*Human Machine Interface*) yang merupakan media interaksi dan operasional berbasis *touch screen*. Dan kedua pemograman *control logic (PLC)*.

Berikut adalah perancangan dan penjelasan dari pembuatan simulator ini

4.1. Perancangan *Hardware* simulator

Berikut ini adalah bentuk desain diagram blok dari simulator yang akan di buat



Gambar 4.1 Rancangan Simulator

Gambar diatas adalah suatu rancangan simulator yang sudah *terintegrasi* antara HMI dan PLC modul. Untuk komunikasi antara 2 alat (*device*) tersebut menggunakan komunikasi *Modbus*, dengan penghantar kabel UTP (*straight connection*) berikut penjelasan device tersebut.

HMI *Touch screen* yang akan digunakan adalah HMI dengan tipe XBTGT2330 dengan ukuran *touch screen* 5,7 inch. Menggunakan *supply* 24VDC 26W. (*Schneider*)



Gambar 4.2 MHI *Touch Screen* XBTGT2330

Dan PLC yang digunakan adalah *Zelio Smart Relay* dengan tipe SR3 B010BD dan modul komunikasi *MODBUS* dengan tipe SR3 MBU01BD.



Gambar 4.3 Modul PLC *Zelio Smart Relay*



Gambar 4.4 Modul *Modbus SR3*

MBU01BD

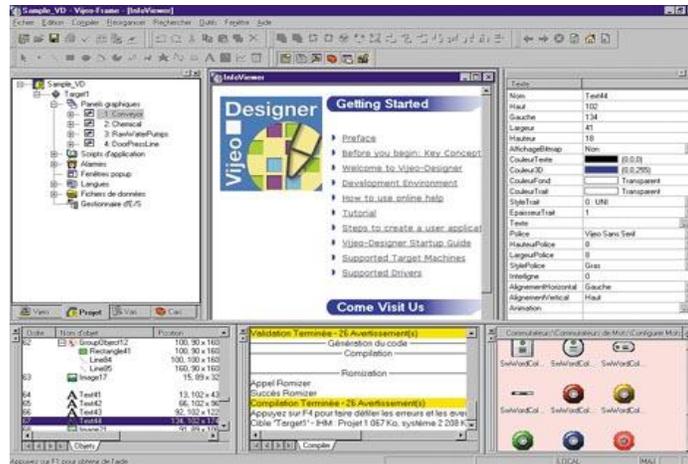
Untuk mengkomunikasikan HMI dengan *Zelio Smart relay* menggunakan kabel UTP CAT 5e (*straight connection*) yang digunakan untuk menghubungkan antara (HMI) XBTGT2330 dengan PLC (*Smart relay*) SR3MBU01BD.



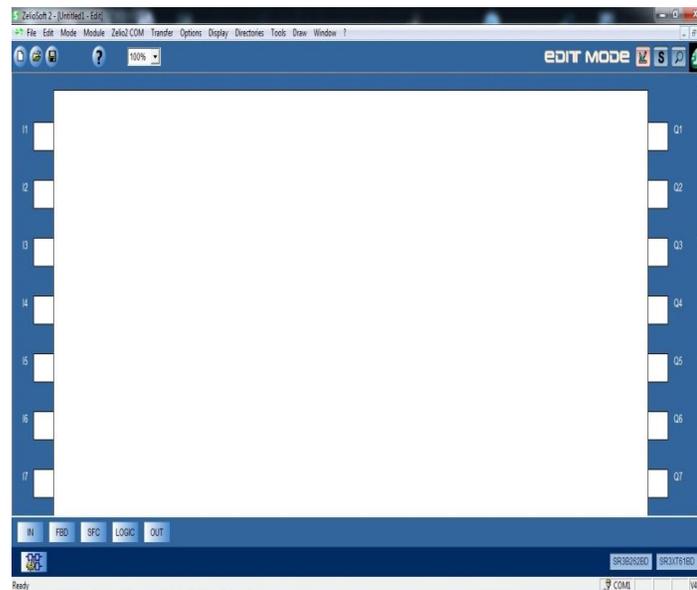
Gambar 4.5 Kabel UTP Cat 5e

Kedua modul tersebut HMI dan Modul PLC, semua nya memerlukan pemrograman, yang disesuaikan dengan kebutuhan sistem kontrol. Dalam kondisi *standart (default)* modul tersebut tidak dapat di gunakan karena masih kosong (*Empty Program*). Untuk pemrograman modul HMI, menggunakan *software Vijeo Designer* untuk *zelio smart relay* (PLC) menggunakan *Zelio Soft 2. (Schneider)*

Berikut adalah tampilan awal pada pemrograman *Software Vijeo Designer* dan *Software Zelio Soft 2*.



Gambar 4.6 Interface software *Vijeo Designer 6.2*



Gambar 4.7 Interface software *Zelio Soft 2*

Untuk proses *Upload* dan *download* program dari kedua modul tersebut, menggunakan kable khusus, untuk Modul HMI menggunakan kabel XBTZG935, untuk

Modul PLC *zelio smart relay* menggunakan (SR2USB01). Kedua kabel transfer tersebut akan di hubungkan dengan komputer melalui port USB, yang telah di sesuaikan *port COM address* nya.

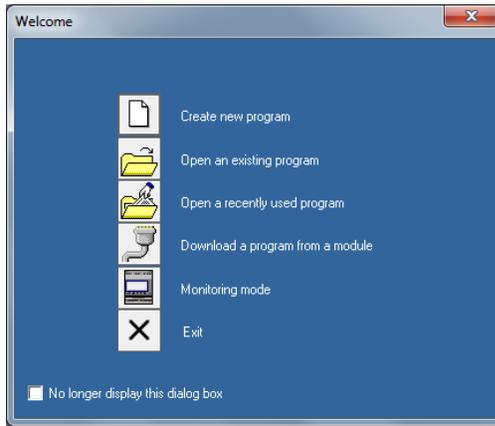


Gambar 4.7 Kabel *transfer* HMI (XBTZG935)
Dan PLC *smart relay* (SR2USB01)

4.2. Pemrograman *Software*

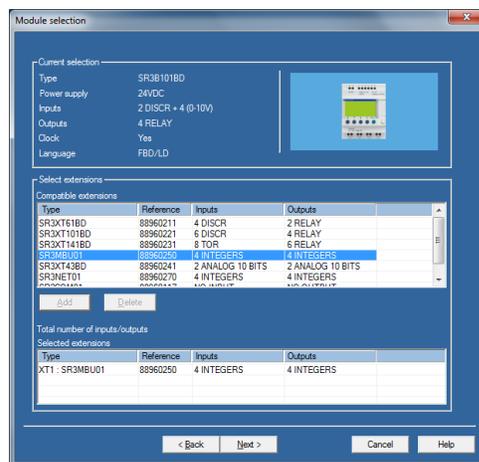
4.2.1. Pemograman *software* PLC (*Zelio Smart relay*) SR3 B010BD

Setelah membuka aplikasi *software zelio soft 2*, dalam jendela ini juga terdapat berbagai mode open program diantaranya, (*Open exisiting program*) buka dari save as komputer. (*Download program from a module*) yaitu membuka program dari modul dengan cara mendownload dari modul. (*Monitoring Module*) membuka program hanya untuk monitor saja. karena ini merupakan awal pemograman kita memilih (*create new program*) yang akan di tunjukkan pada Gambar 4.8.



Gambar 4.8 Jendela open file *zelio soft 2*.

Zelio Soft 2 ini memiliki banyak database module atau seri dari PLC *smart relay* nya. Di sini penulis menggunakan modul SR3 B010BD sebagai modul utama. Kemudian pilih *device* lain yang akan di gabungkan dengan modul utama tersebut. Di sini penulis menggunakan modul SR3 MBU01BD, merupakan modul komunikasi *Modbus*, yang di gunakan untuk komunikasi dengan HMI nantinya. Kemudian pilih (*Add*) tambahkan dan lanjutkan ke jendela berikut nya. Berikut adalah tampilan untuk memilih modul komunikasinya.

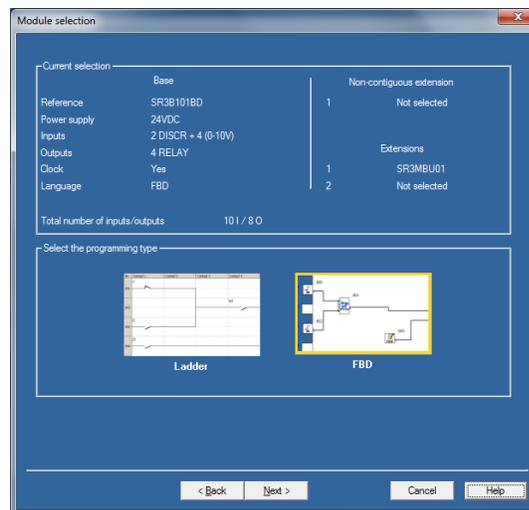


Gambar 4.9 Jendela Pemilihan modul pada *zelio soft 2*.

Setelah melakukan pemilihan module yang digunakan pada gambar 4.9, selanjutnya melakukan pemilihan aplikasi pemrograman. Pada *zelio soft 2* ini terdapat dua

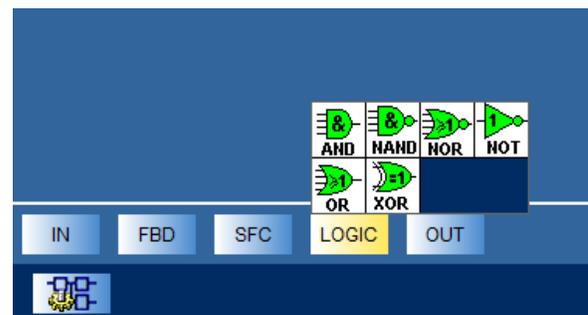
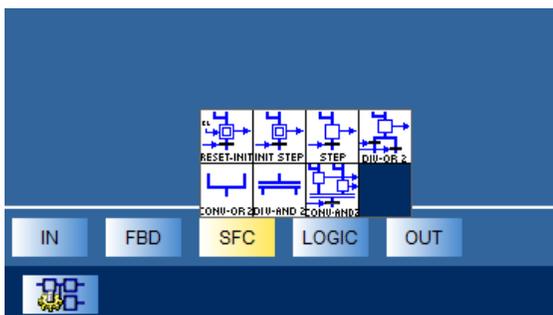
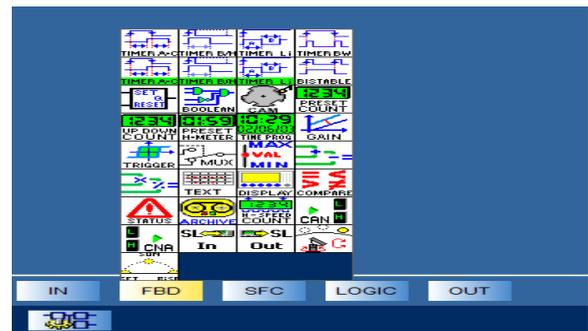
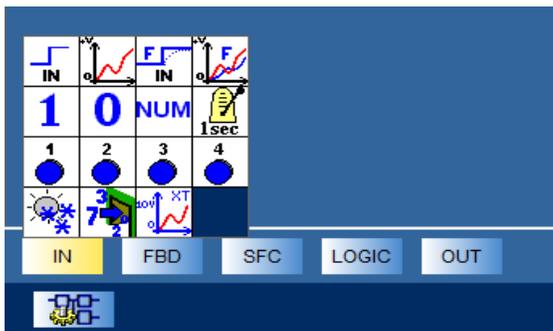
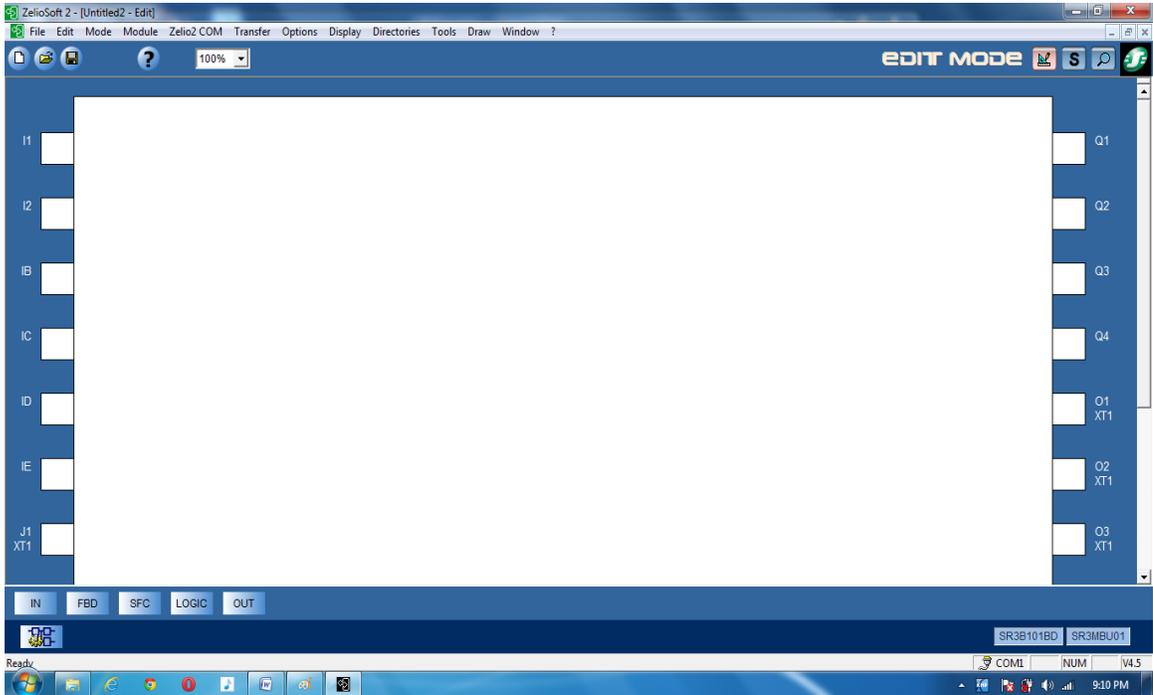
buah cara penulisan program, yaitu dengan *leader diagram* dan *Function Block Diagram* (FBD). Kedua bahasa pemrograman tersebut dapat digunakan. Pada Leader diagram prinsip dasar pemrogramannya yaitu didominasi dengan kondisi NO (*Normaly Open*), NC (*Normaly Close*). Jika menggunakan FBD disini lebih didominasi penggunaan Gerbang logika (*Logic Boolean*) dan *block-block digital* lainnya seperti SCF (*Sequential Function Chart*). Penulisan dengan FBD memiliki beberapa kelebihan yaitu pemrograman menjadi lebih sederhana, dan lebih mudah di monitor ketika dioperasikan.

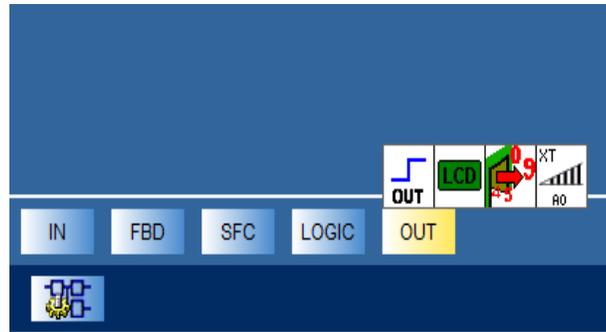
Berikut adalah tampilan modul *selection* pada *zelio soft 2*.



Gambar 4.10 Jendela Pemilihan bahasa pemrograman pada *zelio soft 2*.

Dan berikut adalah tampilan bahasa pemrograman dengan menggunakan program bahasa *Function Block Diagram (FBD)*.





Gambar 4.11 Jendela utama dan menu bahasa pemrograman FBD *zelio soft 2*.

Pada tugas akhir ini penulis memilih simulasi pembuatan mekanisme pengisian air tampungan, dilengkapi dengan *monitoring* level air dan pemilihan operasional secara otomatis atau manual. Pemograman diawali dengan menentukan I/O yang akan digunakan :

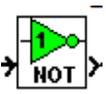
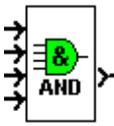
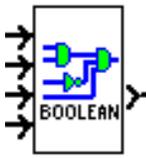
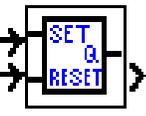
Input

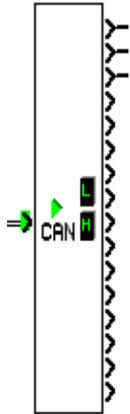
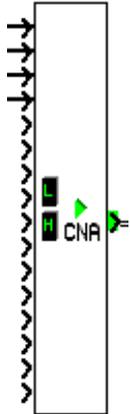
- | | | |
|----|---------------------------------------|-----------------------------------|
| 1. | Level air rendah (<i>Low</i>) | <i>Floating Switch (Hardware)</i> |
| 2. | Level air tinggi (<i>High</i>) | <i>Floating Switch (Hardware)</i> |
| 3. | Kendali <i>Start</i> | <i>Touch Screen</i> |
| 4. | Kendali <i>Stop</i> | <i>Touch Screen</i> |
| 5. | Kendali pemilihan (<i>Selector</i>) | <i>Touch Screen</i> |

Output

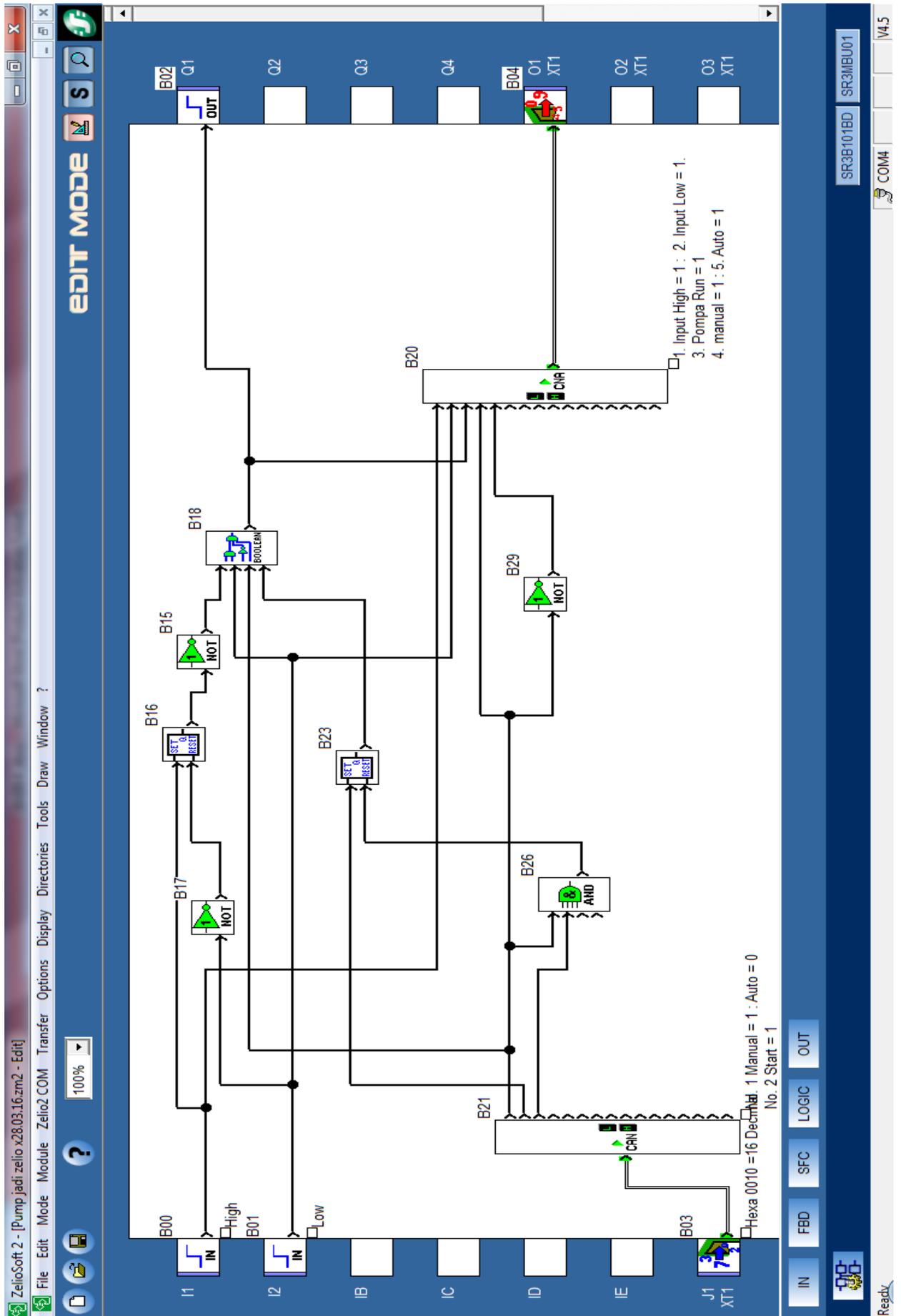
- | | | |
|----|---|----------------------|
| 1. | Indikator rendah (<i>low</i>) | Display monitor |
| 2. | Indikator tinggi (<i>High</i>) | Display monitor |
| 3. | Indikator operasi pompa (<i>Run status</i>) | Display monitor |
| 4. | <i>Output MC (Magnetic Contactor)</i> | <i>Control Motor</i> |

Adapun FBD yang akan di gunakan pada pembuatan simulasi ini :

Fungsi	Simbol	Deskripsi
IN		Fungsi <i>INPUT</i>
<i>LOGIC</i>		Fungsi <i>Logic</i> gerbang NOT
<i>LOGIC</i>		Fungsi <i>Logic</i> gerbang AND
FBD		Fungsi <i>Logic</i> gerbang <i>Boolean</i> . Dimana, Nilai <i>output</i> dapat disesuaikan dengan <i>logic input</i>
FBD		Fungsi <i>logic SET - RESET</i> ,
<i>IN</i>		Input <i>Logic diskrit 1 (High)</i>
<i>OUT</i>		<i>Logic Output</i>
<i>IN</i>		<i>Logic Input Integer</i>

<p><i>OUT</i></p>		<p><i>Logic Output Integer</i></p>
<p>FBD</p>		<p><i>Logic CAN (Word Bit Conversion)</i></p>
<p>FBD</p>		<p><i>Logic CNA (Bit to Word Conversion)</i></p>

Tabel 4.1 Fungsi *Logic* yang digunakan pada pembuatan simulasi.



Gambar 4.12 Pemograman FBD simulator pengisian air dengan zelio soft 2.

Keterangan pada gambar 4.12, Pertama meletakkan *input 1* dan *input 2* pada kolom input dengan cara drag drop dari pilihan menu FBD, I1 sebagai level penuh (High), I2 sebagai level rendah (Low) di dapat dari input sensor *floating witch*. Kemudian memilih *Logic FBD Boolean* dengan kombinasi output :

Input 1	Input 2	Input 3	Input 4	Output
0	0	0	0	1
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	0
0	0	1	1	0
1	0	1	1	1
0	1	1	1	0
1	1	1	1	1

Gambar 4.13 tabel *logic FBD boolean zelio soft 2*.

Pada *zelio soft 2* ini memiliki kelebihan, dimana kita dapat secara bebas menentukan *output* dari sebuah *logic diagram* melalui FBD (*Boolean Function*), sehingga dapat memudahkan dalam pemograman dan mempersingkat pembuatan *logic diagram*. Terlihat pada gambar 4.13, tanda panah merah merupakan output logic yang bernilai diskrit 1 (*High*). Dengan kondisi *input* terlihat pada gambar 4.12.

Input rendah (Low) di hubungkan dengan *Logic boolean input 2*, untuk *input 1 Penuh (High)* menggunakan *logic FBD set-reset*, yang *output* nya di hubungkan dengan *logic boolean input 1*, karena input sensor menggunakan *logic NO (Normaly Open)*, maka pada pemograman ini di gunakan *logic gerbang NOT*, agar input menjadi *NC (Normaly Close)*, sesuai dengan penerapan dari *FBD boolean* pada gambar 4.13 semua sensor level juga di hubungkan dengan *logic (CNA)* agar dapat di baca oleh HMI.

Function block CAN dan *CNA* ini di gunakan untuk mengatur komunikasi antara PLC *zelio smart relay* dengan modul HMI. Modul HMI ini mengirimkan data ke modul PLC *zelio smart relay* melalui modul SR3 MBU01BD modul ini menyediakan empat buah *memory 16 bit*. dimana *port* ini di sebut *input Word J1XT1 – J4XT1 (input Integer)* yang diambil *bit-bit* nya menggunakan fungsi *Word to Bit Conversion (CAN)*. Sedangkan untuk mengirimkan data dari modul PLC ke HMI, modul SR3 MBU01BD ini juga menyediakan 4 buah *output 16 bit O1XT1 – O4XT1 (Output Integer)* .

Dalam rancangan simulasi ini *input integer J1XT1 (Hexa 0010 address)* , CAN di program:

1. Bit 1 digunakan untuk perintah tombol *Auto* dan *Manual*
2. Bit 2 digunakan untuk perintah tombol *Start*
3. Bit 3 digunakan untuk perintah *Stop*

Dan untuk *Output Integer* O1XT1 (*Hexa 0014 address*) CNA di program:

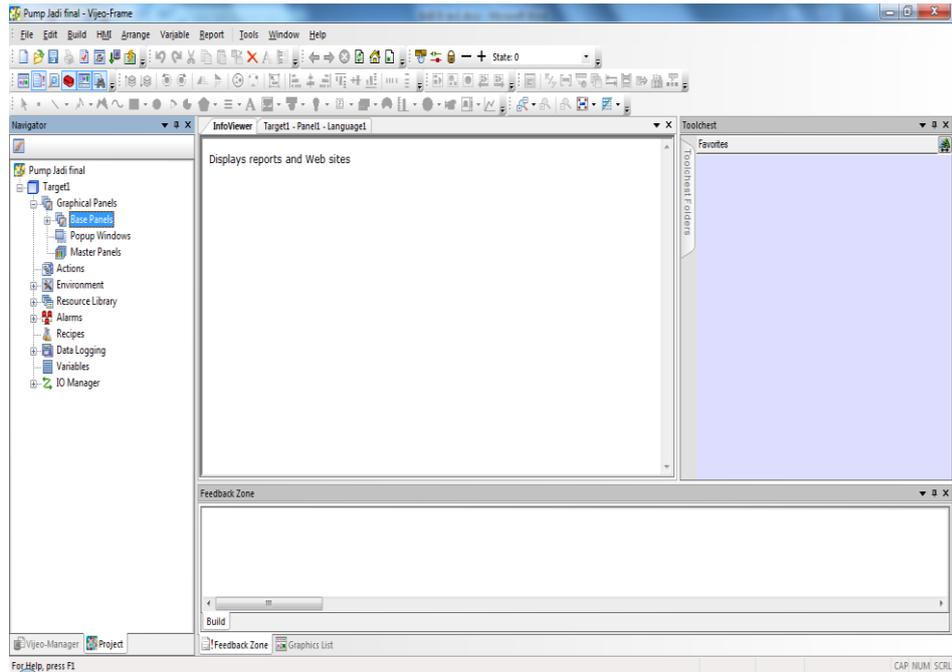
1. Bit 1 digunakan untuk status *Output* penuh/*High I1*
2. Bit 2 digunakan untuk status *Output* rendah/*Low I2*
3. Bit 3 digunakan untuk status *Output* Q1
4. Bit 4 digunakan untuk status *Integer input* Bit1 (*auto* dan manual)

Setelah program pada *zelio soft 2* telah selesai di buat, kita dapat melakukan simulasi pada *software*, dan jika telah seperti yang kita harapkan, maka sekarang program tersebut dapat di *upload* kedalam *hardware zelio smart relay* dengan menggunakan kabel data SR2USB01, yang telah di terangkan di atas.

4.2.2. Pemograman *software* HMI (*Vijeo Designer 6.0*)

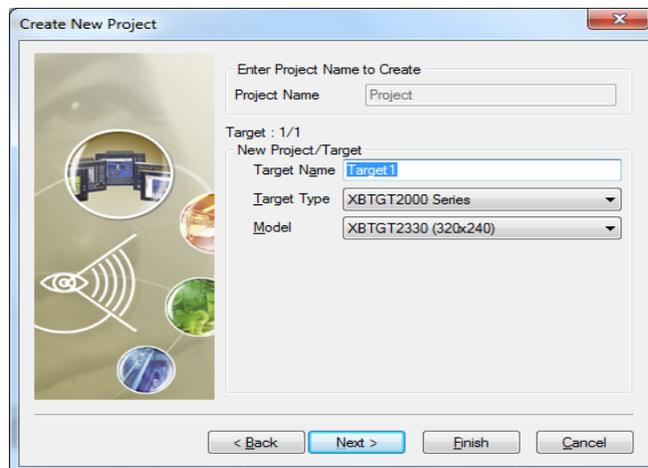
Modul HMI yang penulis gunakan pada tugas akhir ini adalah tipe XBTGT 2330 dengan resolusi (320 x 240). Sebelum di gunakan sama halnya dengan modul PLC *Zelio smart relay*, modul HMI ini juga dilakukan pemograman sesuai dengan kebutuhan dan fungsi *device otomasi* yang kita buat.

Berikut adalah tampilan menu awal pada *Vijeo Desegner 6.0*



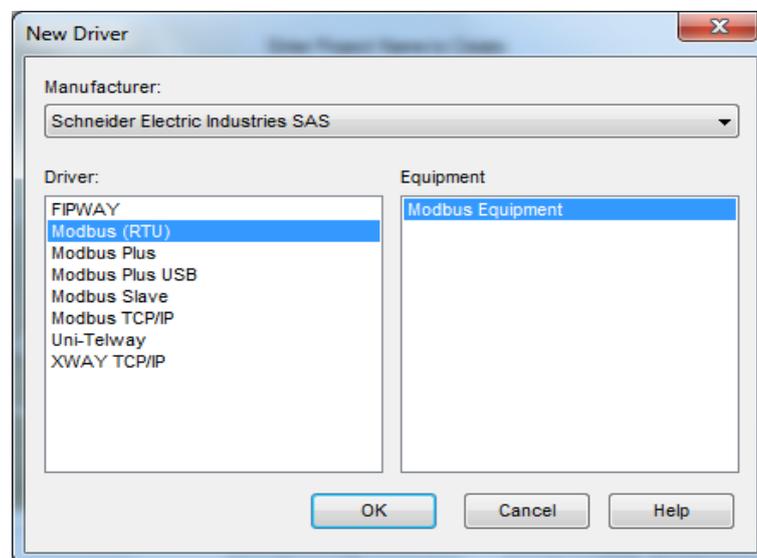
Gambar 4.14 Tampilan dasar *Vijeo Designer 6.0*

Langkah pertama dalam melakukan pemrograman *vijeo designer* ini adalah kita menentukan jenis dan *type* modul *hardware* yang akan di gunakan. Buat nama *project* yang akan kita buat, pilih *target type* : disini penulis menggunakan type XBTGT 2000 series, dengan model XBTGT 2330 (320x240) pada gambar berikut.



Gambar 4.15 Pemilihan *hardware* yang akan di gunakan

Setelah modul *hardware* kita pilih, kemudian kita pilih *type* komunikasi yang akan di gunakan, Pada tugas akhir ini penulis menggunakan komunikasi *Modbus*. Modul HMI XBTGT 2330 ini memiliki media komunikasi data menggunakan, *Ethernet*, *data logging* dan *Modbus*. Karna pada simulasi ini Modul HMI menggunakan jalur RS-485 dan pada PLC *zelio smart relay* menggunakan komunikasi *Modbus*, maka pada jendela pemilihan komunikasi di pilih komunikasi *Modbus RTU*, dimana *type* komunikasi ini terdapat pada modul tambahan *zelio smart relay* (SR3 MBU01BD). Berikut adalah tampilan pada menu pemilihan *type* komunikasi.

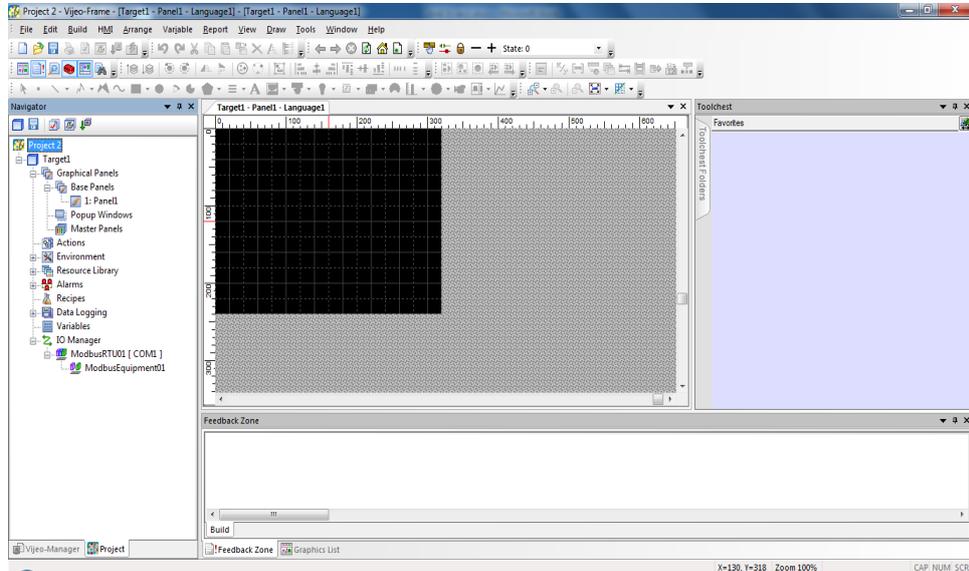


Gambar 4.16 Pemilihan *type* komunikasi data pada *Vijeo Designer 6.0*

Langkah selanjutnya yaitu memasukan semua *variable* yang akan di gunakan baik *input* dan *output* pada HMI, dengan cara kita klik dua kali pada *list Variable*, kemudian pilih *new variable*, dan jendela *new variable* akan tampil, pada pemograman HMI pada tugas akhir ini semua *input*

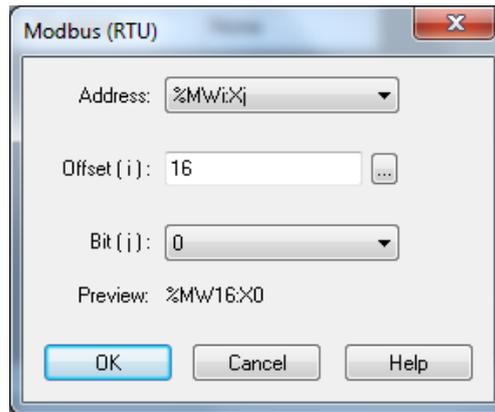
output menggunakan *variable integer*, data yang akan di ambil dan dikirim berasal dari PLC *zelio smart relay*, maka di pilih, data source : *external*.

Berikut adalah tampilan menu pada pemograman *Vijio Desegner 6.0*



Gambar 4.17 Jendela pembuatan *new variable* pada *Vijeo Designer 6.0*

Setelah itu kita memasukan alamat *device* yang akan di gunakan, salah satu contoh *variable input* tombol pemilihan *auto manual* pada gambar berikut.



Gambar 4.18 Menentukan *device address*

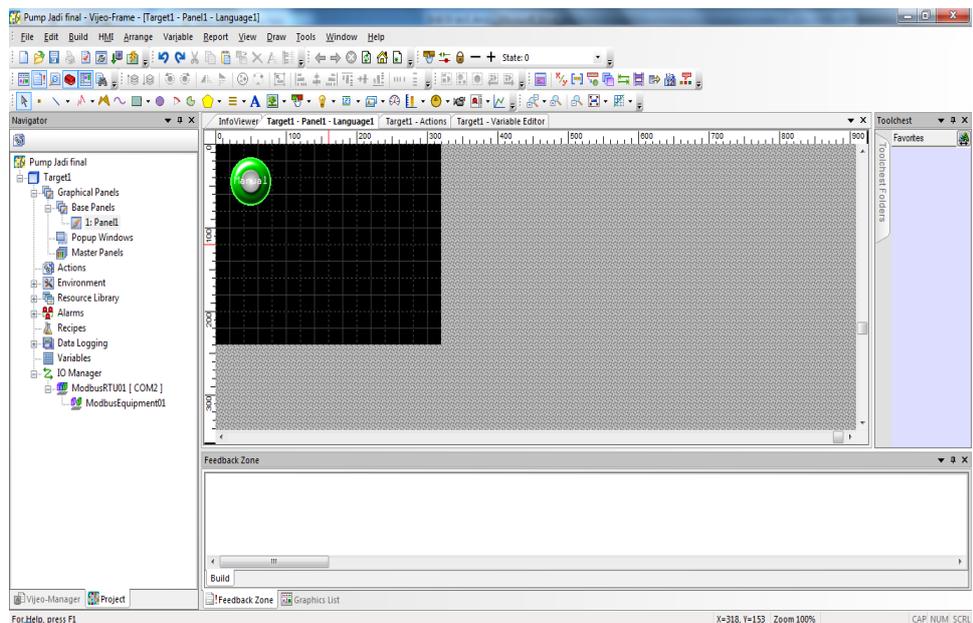
Karena menggunakan alamat *address external*, kita memilih memori *address type* %MWiXi. Kemudian kita mengisikan *offset*. Dalam pemilihan *offset* ini melihat dari *device external* yang di gunakan. pada PLC *zelio smart relay* seperti diuraikan diatas untuk *input* dan *output* harus di *konversi* kedalam desimal.

1. *Input Integer* J1XT1 0010 Hexa = 16 *Decimal*
2. *Output Integer* O1XT1 0014 Hexa = 20 *Decimal*

Kemudian pada kolom *offset input* di masukan angka 16, dan pada kolom *offset variable output* dimasukan nilai 20 kemudian untuk *output* salah satu contoh, O1XT1 (*Hexa 0014 address*) begitupun selanjutnya untuk *input integer* dan *output integer* yang lain.

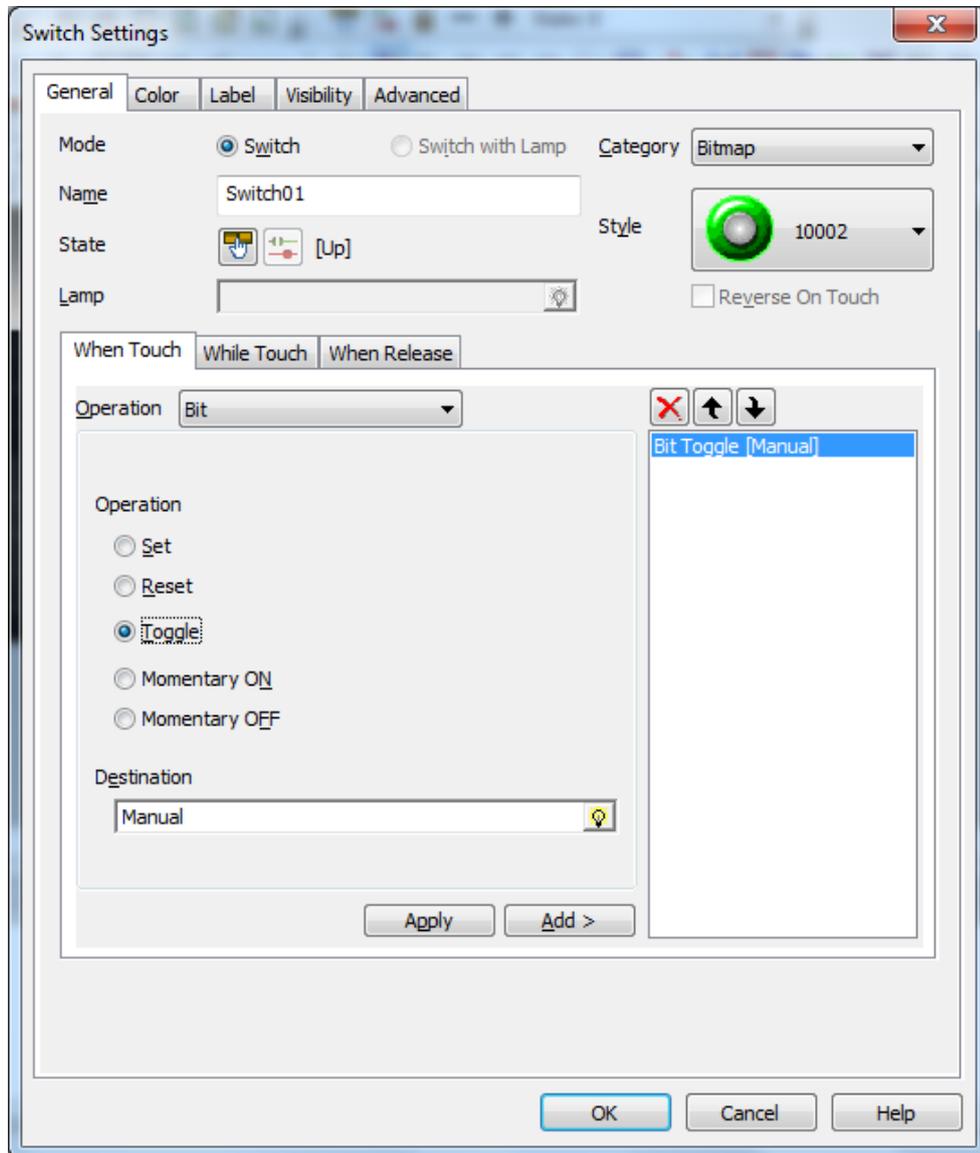
Setelah semua *variable input* dan *output* sudah di buat, selanjutnya yaitu meakukan desain tampilan pada panel, dalam melakukan desain dapat menggunakan beberapa *template* yang telah di sediakan dalam *software*

vijeo designer ini. Sebagai contoh untuk input dapat kita pilih gambar *push button* pada gambar berikut,

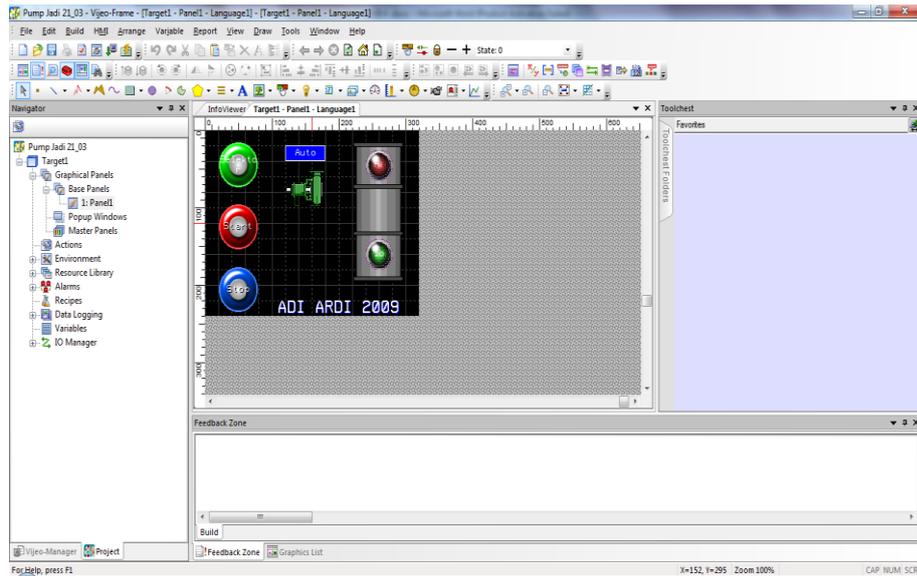


Gambar 4.19 Jendela desain panel

Setelah gambar atau *template hardware* di pilih, selanjutnya kita akan memasukan fungsi *variable* dari gambar push button tersebut, dengan cara klik dua kali pada *template* gambar dan akan muncul jendela *variable setting*, gambar 4.19, pada *kolom destination* kita pilih *variable integer* untuk fungsi tombol tersebut. Selanjutnya menentukan fungsi operasional tombol tersebut, apakah sebagai *set*, *reset*, *toggle* atau *momentary on / off*, dan terakhir kita klik ok. Begitupun selanjutnya untuk gambar-gambar *template* yang lain nya. Setelah selesai maka akan tampil seperti pada gambar 4.20.



Gambar 4.20 Variable setting



Gambar 4.21 Gambar pemrograman tampilan panel

Selanjutnya setelah pemrograman HMI telah selesai dengan yang di harapkan pada gambar 4.21 langkah berikutnya adalah melakukan *upload* program. Dengan cara menyambungkan kabel data antara modul HMI dengan komputer, menggunakan kabel data XBTZG935. Dan lakukan proses *Upload* program.

Setelah semua modul *hardware* telah di program pada PLC *smart relay* dan HMI *touch sreen*, maka kedua modul dapat dihubungkan dengan menggunakan kabel UPT Cat 5e, dan modul PLC di kondisikan kedalam mode *RUN*, maka sistem pengisian penampungan air sekarang dapat berkerja.

Adapun proses kerja dari simulator ini adalah jika volume air di dalam penampungan turun atau rendah maka pompa akan menyala dan mengisi penampungan hingga level air *high* tersentuh, kemudian pompa akan berhenti mengisi, proses selanjutnya yaitu pompa tidak akan berkerja

kembali hingga level air rendah, sensor level aktif atau bekerja. Dalam proses ini juga di berikan dua pilihan mode operasional, yaitu *auto* dan manual, untuk mode *auto* seperti diterangkan diatas, untuk manual yaitu pompa dapat berkerja dan di hentikan setiap saat.