

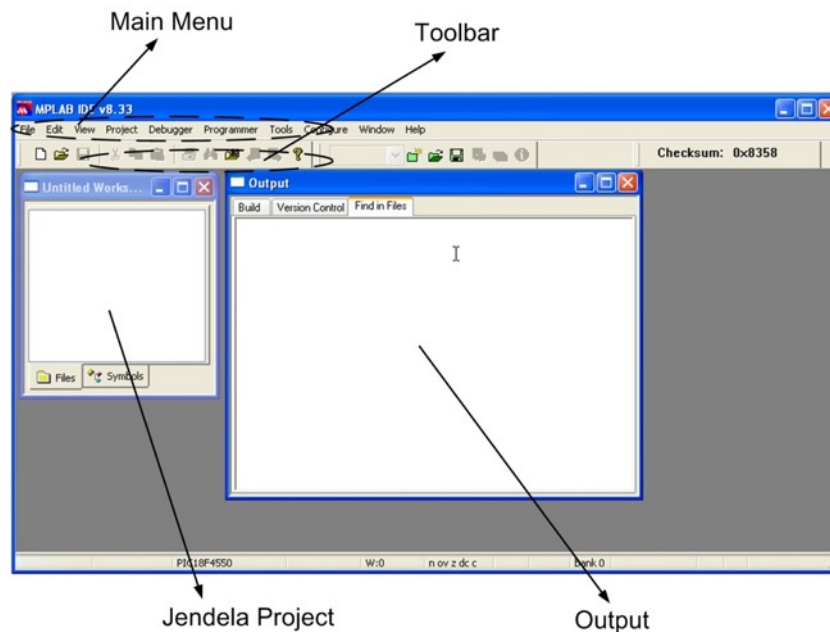
BAB II MPLAB IDE dan PEMROGRAMAN C UNTUK PIC 18F4550

MPLAB IDE merupakan tool software dari Microchip dan dapat didapatkan secara gratis dari situs resmi Microchip di www.microchip.com. (tersedia juga di dalam CD bawaan buku ini). Software MPLAB IDE mendukung semua sistem operasi Windows. Bahasa pemrograman chip mikrokontroler secara default adalah *assembler* standar Microchip. Namun dalam perkembangannya, banyak vendor ketiga yang mengembangkan bahasa pemrograman sendiri seperti *bahasa basic* dan *bahasa C*.

MPLAB IDE mendukung pemrograman semua produk mikrokontroler dari generasi pertama sampai 32 bit. Hanya saja ketika instalasi pertama dilakukan, MPLAB IDE tidak mendukung semua bahasa pemrograman. Agar dapat mendukung bahasa pemrograman yang lain, maka perlu diinstall lagi software kompilasi bahasa secara terpisah. Beberapa pihak/perusahaan pengembang bahasa C dan basic yang mendukung produk Microchip misalnya Oshonsoft, Me-Lab, Hi-Tech C. Mereka menawarkan kemudahan dalam memprogram PIC karena rutin-rutin pemrograman yang sudah mereka sediakan, dan pemakai lebih enak dalam mengkonfigurasi PIC. Namun di samping kemudahan itu, kita harus membeli dari software/source code yang mereka sediakan.

MEMULAI MPLAB IDE

Pada saat pertama kali MPLAB IDE dijalankan, akan muncul jendela seperti pada Gambar 2.1. Pada jendela tersebut, dibagi menjadi jendela Main menu, tool Bar, dan Project. Masih ada beberapa tool lain semuanya dapat diatur dalam pilihan menu View.

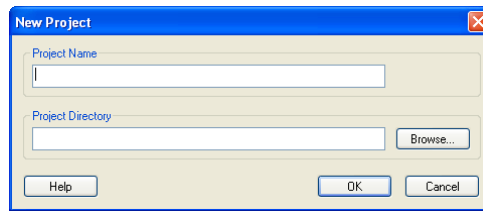


Gambar 2.1. Tampilan MPLAB IDE pertama kali.

MPLAB bisa dijalankan untuk sebuah file *.asm yang tidak memerlukan pendefinisian sebuah project yang komplit. Namun apabila kita bekerja dengan menggunakan lebih dari satu buah file, sebuah project mutlak dibuat. Hal tersebut dikarenakan dalam sebuah project, MPLAB mampu *handle* beberapa file yang bertautan, berbeda jika kita hanya menggunakan sebuah file *.asm saja.

Pembuatan sebuah project baru dapat dilakukan dengan cara sebagai berikut:

Dari MPLAB IDE Main menu pilih **Project> New...**
Sebuah jendela dialog akan muncul seperti Gambar 2.2.



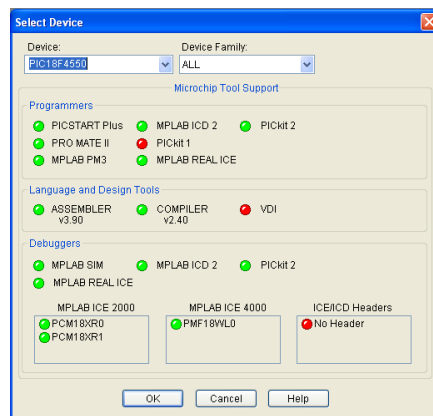
Gambar 2.2 Jendela New Project

Sekarang kita tinggal isi nama project yang akan kita buat pada “Project Name” dan klik tombol “Browse” untuk memilih dimana kita akan membuat direktori project yang kita buat. Kemudian klik OK.

Memilih Jenis Mikrokontroler Yang Digunakan

MPLAB IDE mendukung semua produk mikrokontroler mulai dari 8 bit sampai 32 bit. Untuk project dalam buku ini, kita menggunakan PIC18F4550. Cara memilih mikrokontroler yang kita gunakan yaitu dengan cara:

Pilih **Configure>Select Device** lalu pilih **PIC18F4550** di daftar device lalu klik OK.



Gambar 2.3. Pilihan Mikrokontroler yang akan digunakan.

Memilih Jenis Kompiler Dan Bahasa Pemrograman Yang Digunakan

Jenis kompiler menentukan bahasa pemrograman yang akan kita gunakan. Jika kita menggunakan bahasa assembler, maka kita tidak perlu menginstall kompiler lain semisal C-18, HiTech, dan lain sebagainya. Hal tersebut karena secara default MPLAB IDE menggunakan bahasa assembler/rakitan standar Microchip. Bahasa yang digunakan dalam buku ini adalah menggunakan C18 buatan dari Microchip. Kita dapat mengunduhnya dari situs Microchip. (sudah ada di dalam CD). Versi gratis (Lite edition) dapat kita unduh secara cuma-cuma dengan fitur yang komplit, namun setelah 60 hari ada beberapa fitur yang hilang. Sayangnya untuk upgrade ke full vesion, kita harus membeli dari Microchip. C18 merupakan software yang dibuat oleh Microchip untuk mengkompile listing program (bahasa C) yang telah kita buat agar ditransfer ke bentuk heksadesimal (.HEX). Bahasa C18 mendukung semua mikrokontroler PIC seri 18 dan 24.

C-18 : Bahasa Dan Kompiler C Dari Microchip

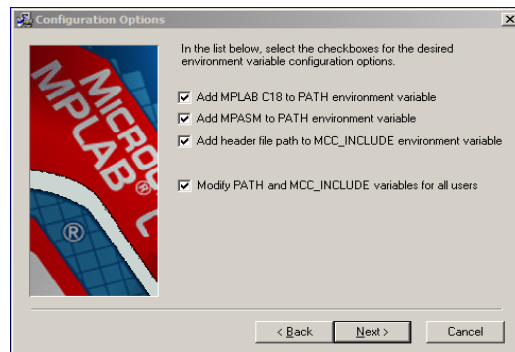
Bahasa C yang digunakan pada pemrograman mikrokontroler pada dasarnya mempunyai aturan penulisan yang sama dengan standar bahasa C secara universal (standard ANSI). Bedanya, ketika memprogram sebuah sistem mikrokontroler dengan bahasa C kita harus sedikit banyak mengetahui datasheet chip yang bersangkutan termasuk jenis register dan kegunaannya, serta bagaimana sistem komunikasi dalam keseluruhan sirkuit akan berhubungan dengan mikrokontroler.

Bahasa C mempunyai beberapa keunggulan dibandingkan dengan bahasa rakitan/assembly. Diantaranya :

1. Sintak perintah bahasa C lebih universal dibandingkan dengan bahasa assembly yang belum tentu sama antar pabrikan chip.
2. Inisialisasi suatu algoritma program dengan sintak bahasa C akan mempunyai kode yang lebih simpel daripada assembly.
3. Jika kita menggunakan fasilitas interface USB dan ethernet, sampel program yang disediakan oleh Microchip sebagian besar menggunakan bahasa C. Dengan adanya sampel programming pada komunikasi USB, kita hanya perlu mengedit dan mempelajari sintak kode yang telah ada. Selanjutnya tinggal memikirkan bagaimana fungsionalitas dari peripheral yang kita rancang

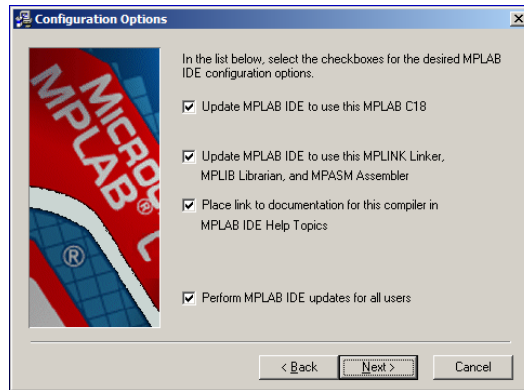
Di sisi lain, kerugian bahasa C adalah hasil kompilasi yang ukurannya besar sehingga memerlukan memori lebih banyak daripada menggunakan bahasa assembly. Bahasa C yang digunakan untuk pemrograman PIC sendiri dapat berasal dari Microchip maupun buatan pengembang software ketiga.

Instalasi C18 merupakan syarat mutlak agar MPLAB dapat mengenali dan akhirnya mengkompilasi file yang notabene berbahasa C (C ANSI Standard). C18 diinstall setelah kita menginstall MPLAB IDE. Agar dapat terintegrasi secara otomatis dalam MPLAB, maka sewaktu menginstall C18 kita perlu mengeset beberapa parameter yaitu dengan mencentang beberapa pilihan seperti pada Gambar 2.4.



Gambar 2.4. Pilihan settingan C18 pada waktu kita melakukan instalasi pertama kali (setelah MPLAB IDE terinstall).

Setelah tombol “Next”, jendela selanjutnya juga menanyakan apakah kita akan menambahkan bahasa dan kompilasi C18 ke dalam lingkungan MPLAB IDE (Gambar 2.5), pilih ya dengan mencentang semua pilihan yang ada.



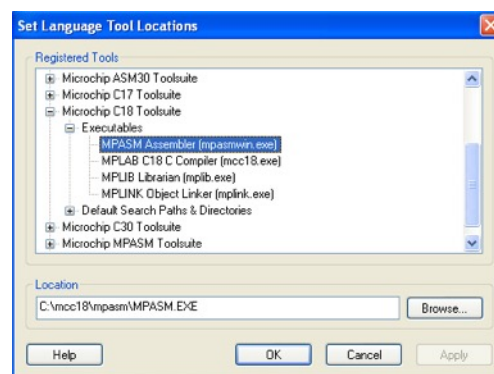
Gambar 2.5. Pilihan agar C18 terintegrasi ke MPLAB IDE.

Setelah semua proses instalasi terlewati C18 akan membuat file rujukan standar di directory C:\MCC18 yang terdiri dari beberapa subfolder lagi yaitu:

- *bin* : berisi file eksekusi untuk kompiler dan linker.
- *cpp* : berisi source code MPLAB C18 C preprosesor
- *doc* : berisi dokumentasi dari MPLAB18
- *example* :berisi contoh-contoh aplikasi menggunakan c18.
- *h* :berisi header file untuk standar librari bahasa C dan presesor librari mirochip MCU.
- *lib*: berisi librari C standar (*clib.lib* atau *clib_e.lib*) , librari untuk jenis prosesor (*p18xxx.lib* atau *p18xxx_e.lib*) serta modul untuk start up (*c018.o,c_018i.o,c018i_e.o,c018iz.o, c018iz_e.o*)
- *lkr*: berisi file linker, untuk menentukan jenis PIC yang digunakan.
- *mpasm*: berisi perintah-perintah dari MPASM assembler.
- *src* : mengandung source code dalam format C dan assembly.

Pengaturan MPLAB IDE

Kembali ke MPLAB IDE, karena yang akan kita bahas kali ini pemrograman dengan bahasa C, kita perlu mengeset jenis kompiler yang akan digunakan oleh MPLAB IDE. Caranya yaitu: Klik **Project>Set Language Tool Locations** lalu untuk memilih bahasa di bawah Microchip C18 C Compiler pastikan pilihan masing-masing *tool* sesuai dengan standard C18 yaitu di C:\mcc18.



Gambar 2.6. Pemilihan kompiler pada MPLAB IDE diset pada Microchip C18 Toolsuite

Dengan memilih “Microchip C18 Toolsuite” maka kompiler yang akan kita gunakan sudah mengarah ke C18. Dengan asumsi instalasi C18 ke C:\mcc18 maka pengaturan pengarah

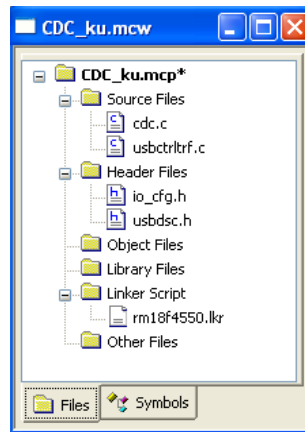
bahasa/kompiler (Gambar 2.6) adalah sebagai berikut :

- MPASM Assembler (mpasm.exe) : C:\mcc18\mpasm\MPASM.EXE
- MPLINK Object Linker (mplink.exe) : C:\mcc18\bin\mplink.exe
- MPLAB C18 C (mcc18.exe) : C:\mcc18\bin\mcc18.exe
- MPLIB Librarian (mplib.exe) : C:\mcc18\bin\mplib.exe

Apabila setting bahasa/kompiler sudah sesuai dengan kriteria di atas, klik “OK” dan MPLAB IDE akan secara langsung mengarahkan kompiler sesuai dengan pemilihan directory yang telah kita tentukan.

Memulai Membuat Project

Setelah pemilihan jenis mikrokontroler, maka kita siap untuk menuliskan kode-kode program yang kita inginkan. Sebuah project akan terdiri satu, dua, atau lebih file source code, header, linker, ataupun yang merupakan listing program yang kita buat (lihat Gambar 2.7). Kita bisa membuat file yang baru, atau sekedar menambah file yang sudah ada ke dalam project.



Gambar 2.7. Susunan jendela dan file dalam sebuah project

Untuk membuat file yang baru, pilihlah pada menu MPLAB IDE :

File > New >... Lalu ketikkan kode program yang akan kita buat di jendela editor. Setelah selesai disimpan dengan ekstensi .c, jika kita menginginkan file tersebut sebagai source Files, sedangkan Header Files berekstensi .h.

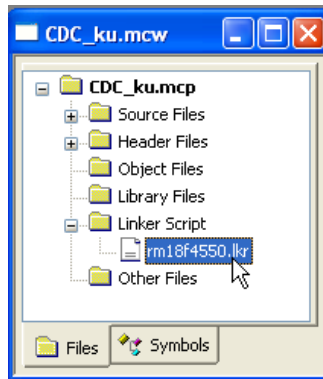
Setelah menuliskan kode program, kita simpan source code program yang telah kita buat dengan cara memilih : **File > Save ...**

dan simpan di satu direktori yang sama dengan direktori project yang kita buat dengan ekstensi “.c”. Sekali tersimpan, maka editor akan secara otomatis sensitif terhadap kata-kata yang mengandung perintah dalam bahasa C.

Penambahan file ke dalam projek dapat dilakukan dengan : **Project > Add Files to Project...** lalu pilih file yang akan ditambahkan ke project. Secara otomatis MPLAB IDE akan memfilter ekstensi file yang kita pilih untuk ditambahkan ke dalam directory yang bersesuaian.

Memilih Script Linker

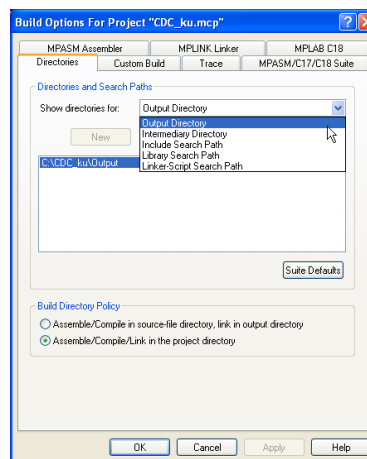
Fungsi linker yaitu sebagai penghubung keluaran dari kompiler dan menghubungkannya dengan bagian yang terkait dengan mikrokontroler seperti RAM, alamat memori, dan Port. Skrip linker terdiri atas 3 bagian file yaitu startup, file librari, dan alokasi memori. Penambahan file linker ini dapat dilakukan dengan mengklik kanan pada **Linker script** dan pilih **Add Files** atau dengan cara menambahkan file dengan ekstensi *.lkr ke dalam project.



Gambar 2.8. File Linker

“SET PATHS”

Sebelum menyelesaikan sebuah project, kita wajib menentukan beberapa path agar MPLAB IDE mampu mencari file entah itu linker, source, ataupun header yang saling berkaitan. Pengaturan path MPLAB IDE dilakukan dengan cara **Klik Project > Build Options ... > Project** untuk membuka kotak dialog Build Option. Pilih tab **General** dan tentukan letak folder untuk Output, Intermediate, dan seterusnya seperti Gambar 2.9.



Gambar 2.9. Kotak dialog pemilihan Path.

Output Directory merupakan tempat untuk meletakkan keluaran file berupa *.hex*, *.cof* dan *.map*. *Intermediary directory* tempat untuk meletakkan file keluaran object dari project. Karena keduanya merupakan keluaran dari kompiler C18, ada baiknya Output dan Intermediary dijadikan satu folder dalam satu project. Sedangkan *Include search Path* merupakan tempat dimana file header dan penunjuk file dalam program yang kita buat diarahkan. *Library Search Path* merupakan tempat beberapa rutin diletakkan. Begitu juga dengan *Linker-Script Search Path* akan mengacu dimana file linker diletakkan.

“MAKE” DAN “BUILD THE PROJECT”

Setelah semua program dan path kita tentukan, kita dapat membuat project kita ke file *.hex* dengan cara pilih **Project > Build All...** untuk membangun project yang kita buat. Alternatif yang lain, kita juga bisa menggunakan **Project > Make....**

Dengan menggunakan *Build All* semua file kecuali file source akan di hapus, dan akan dibangun

BAB III DOWNLOADER DAN BOOTLOADER

MICROCHIP 18F DAN CARA MENDOWNLOAD PROGRAM

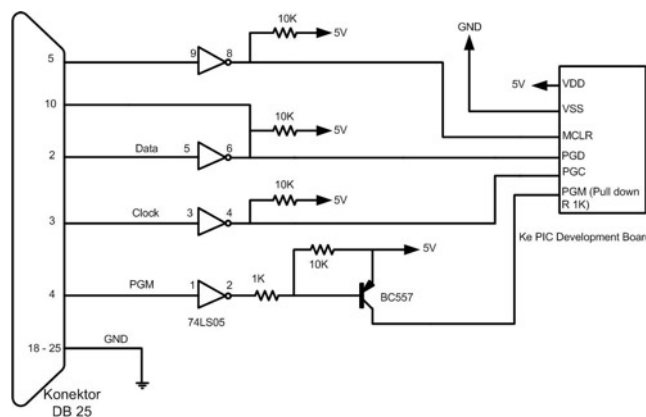
*.HEX

Secara umum ada dua metode untuk mengupload/menanam program .HEX ke mikrokontroler jenis PIC. Pertama metode tegangan rendah. Metode ini menggunakan tiga kabel komunikasi selain MCLR, VDD dan VSS yaitu **PGC (Clock)**, **PGD (Data)**, dan **PGM**. Dinamakan tegangan rendah karena tegangan MCLR pada waktu beroperasi sebesar hanya 5 volt dan bisa diparalel dengan tegangan VDD. Metode kedua adalah tegangan tinggi, yaitu dengan hanya menggunakan 2 kabel jalur data **PGD** dan **PGC** (selain MCLR, VDD dan VSS) dengan tegangan MCLR sebesar 13 volt.

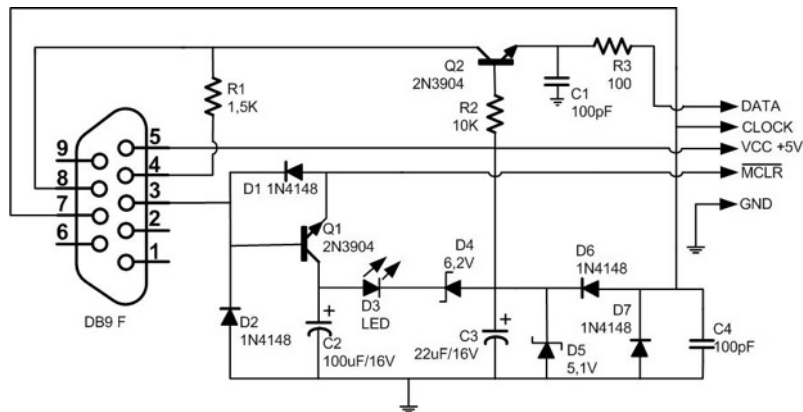
Kedua metode tersebut sama-sama dalam satu fasilitas ICSP (*InLine Circuit System Programming*). Keuntungan ICSP adalah programming secara langsung dalam rangkaian itu juga, tanpa harus melepas dan memasang IC setiap kali akan *update firmware*. Ada banyak downloader yang di pasaran, beberapa diantaranya adalah PICPgm dan WinPic800.

1. PICPgm

PICPgm merupakan programmer untuk microchip yang sudah lama ada, dengan antarmuka ke PIC menggunakan parallel port (DB25). Gambar rangkaian untuk downloader ini ditunjukkan pada Gambar 3.1. Selain parallel port DB25, ada seri DB9 JDM Programmer yang kompatibel dengan software PICPgm. Bahkan ada juga versi pengembangan dengan IC FTDI paralel to USB, sehingga antarmuka yang disuguhkan sudah berupa USB.

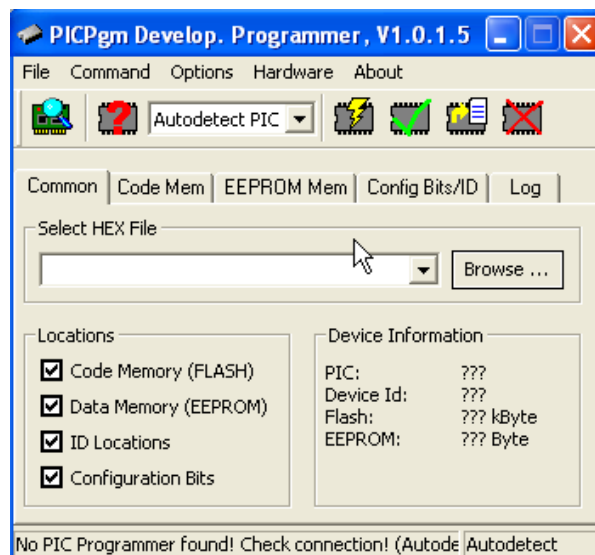


Gambar 3.1. Rangkaian Hardware untuk PICPgm dengan konektor LPT1 (DB25)



Gambar 3.2. Rangkaian Hardware untuk PICPgm dengan konektor COM (DB9)

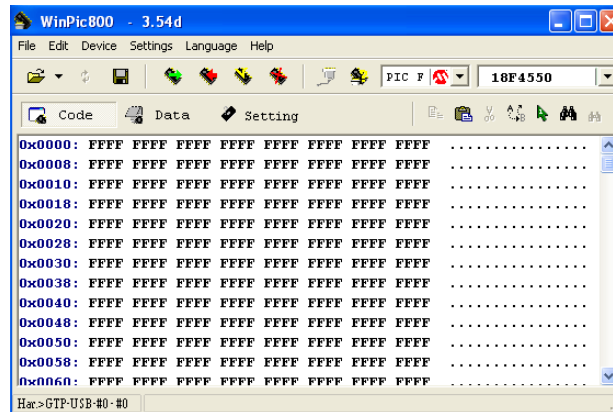
Software sisi PC menggunakan **PICPgm Development programmer**, dengan pilihan antarmuka bisa DB25 ataupun DB9. Keterangan lebih lanjut dapat dilihat di website PICPgm di http://members.aon.at/electronics/pic/picpgm/_main.html.



Gambar 3.3. Tampilan GUI (Graphical User Interface) PICPgm pada PC

2. WinPic800

Rangkaian WinPic800 dapat kita dapatkan secara cuma-cuma di situs www.winpic800.com. Seri WinPic800 sampai saat ini sudah bisa mendukung kebanyakan mikrokontroler PIC dari 8 bit sampai 32 bit, mikrokontroler AVR dari ATMEL, serta beberapa EEPROM. Namun untuk mendapatkannya ini kita harus membeli firmware 18F2550. Metode yang digunakan dalam memrogram PIC berbeda dengan PICPgm yaitu dengan metode tegangan tinggi. Keuntungan dari WinPic800 adalah antarmuka yang sudah mendukung USB dan dukungan terhadap jenis PIC yang luas, mulai dari seri 16xx sampai beberapa seri 32xx.

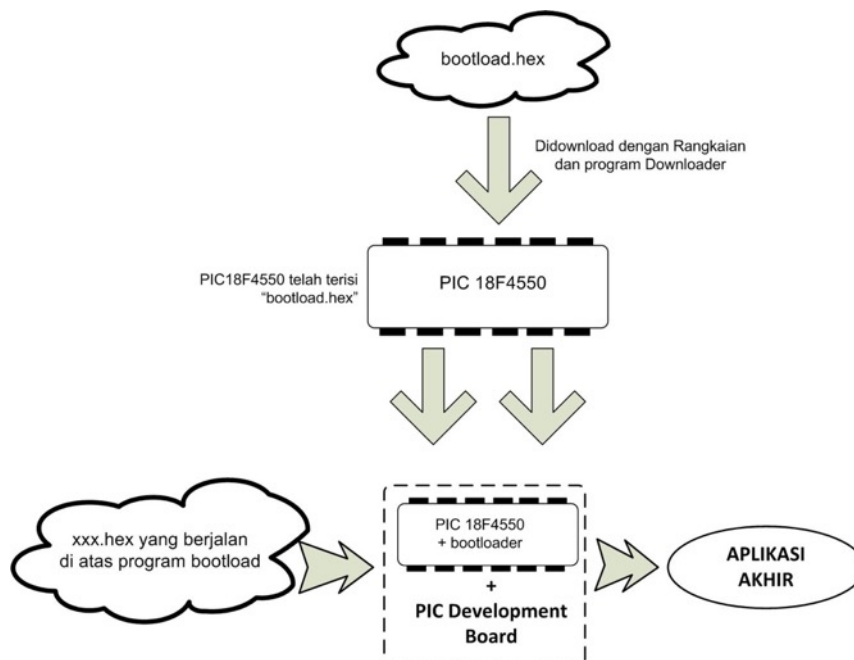


Gambar 3.4. Tampilan GUI WinPic800

Downloader tersebut di atas hanyalah salah satu dari beberapa downloader yang ada untuk memprogram PIC dengan program yang telah kita buat. Beberapa alternatif downloader juga banyak ditawarkan oleh vendor penyedia downloader untuk PIC. Pemilihan dan pertimbangan harga, komponen, dan kenyamanan pengguna selebihnya merupakan pilihan masing-masing individu.

MICROCHIP MCHPFSUSB BOOTLOADER

Bootloader berfungsi mengupload firmware ke device. Berbeda dengan downloader, bootloader USB PIC adalah program kecil yang menempati memori program di alamat 0 sampai 2048 byte pertama pada target PIC. Bootloader berjalan pertama kali pada waktu booting. Kondisi ini tercapai pada saat mikroprosesor reset. Bootloader merupakan firmware yang tetap tinggal di PIC, dan pada seri 18F4550 memungkinkan kita untuk mengupload firmware aplikasi kita yang baru ke chip melalui koneksi USB. Ilustrasinya seperti pada Gambar 3.5.



Gambar 3.5. Proses pemrograman memanfaatkan bootloader

Mula-mula PIC diisi program dengan *bootload.hex* (*bootload.hex* hanyalah nama inisial saja). Setelah itu, dengan tambahan saklar push button untuk RESET dan BOOT yang terhubung dengan Port B4 (lihat Gambar 3.8), kita bisa memanfaatkan PIC untuk mendownload program .HEX tanpa menggunakan downloader lagi. Dengan demikian akan mempermudah kita dalam mengupdate firmware PIC, tanpa mengubah koneksi dan konfigurasi hardware.

Microchip sendiri telah menyediakan program bootload ini lengkap dengan *source codenya*. Metode bootload yang disediakan oleh Microchip ada dua macam yaitu kelas HID dan kelas khusus (custom class). Kelas HID membutuhkan instalasi .NET untuk aplikasi sisi Host, namun dapat langsung mendeteksi keberadaan hardware secara otomatis. Sedangkan metode kelas khusus (*Custom Class*) USB menggunakan driver yang disediakan oleh Microchip. Bootload yang digunakan dalam buku ini menggunakan kelas khusus Microchip, bukan HID.

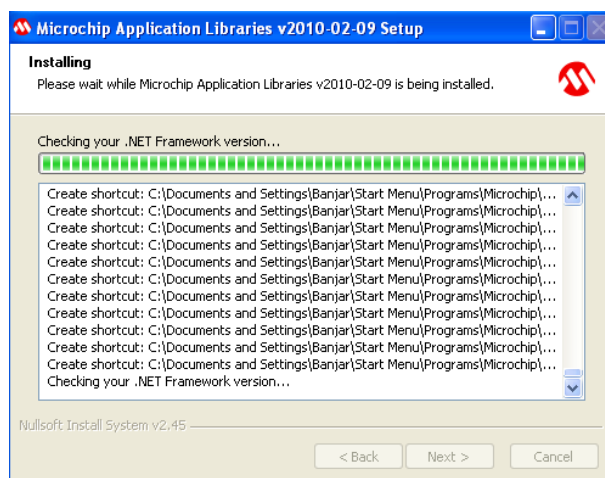
Untuk mengimplementasikan bootload, dan memungkinkan user mengupload program *.HEX ke PIC ada dua hal yang harus dipenuhi. Pertama di sisi PIC18F4550, harus sudah terisi firmware untuk bootload. Kedua, dari sisi PC harus ada software untuk mengupload kode yang telah kita buat. Sehingga jika dilihat dari sisi device, langkah pertama adalah kita mengupload program “bootload.hex” ke dalam PIC18F4550 melalui WinPic800 atau PICPgm.

Dengan adanya firmware “bootload”, PIC sudah siap untuk digunakan untuk diisi firmware baru melalui koneksi USB. Software di Windows juga bukan lagi program downloader (PICPgm atau WinPic800) melainkan **PDFSUSB.exe** pada folder : C:\Microchip Solutions\USB Tools\PDFsusb (setelah MCHP_App_Libvxx_Installer terinstall).

Dengan USB PIC Bootloader yang sudah tertanam di PIC, ada 2 mode operasi yaitu **mode firmware update** dan **mode user**. Pada posisi Mode Update Firmware, USB PIC Bootloader memanfaatkan koneksi USB antara Mikrokontroler dengan PC dan meload kode aplikasi baru. Setelah selesai memprogram, bootloader dipindah ke mode user, prosesor reset dan mulai menjalankan kode program yang baru. Pada posisi **mode user**, bootloader memetakan kembali alamat interrupt vektor aplikasi firmware yang sedang berjalan.

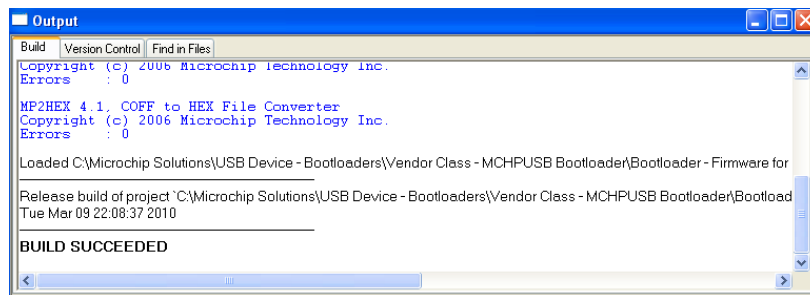
Bagaimana praktik nyata bootload menggunakan PIC18F4550? Pertama-tama kita harus mendownload program **MCHP_App_Lib v2010_02_09_Installer** yang dapat diunduh dari situs Microchip : www.microchip.com bagian *USB development Tool and Program*. Perhatikan, barangkali versi librari yang disediakan oleh Microchip bisa berbeda, namun pada akhirnya tetap sama – sama menggunakan folder yang sama di C:\Microchip Solutions. Langkah-langkah dalam membuat sebuah file bootload yang siap diprogram ke PIC18F4550 adalah sebagai berikut:

1. Pertama, Install software **MCHP_App_Lib v2010_02_09_Installer** secara *default*, sehingga kita mempunyai directory C:\Microchip Solutions.(Gambar 3.6)



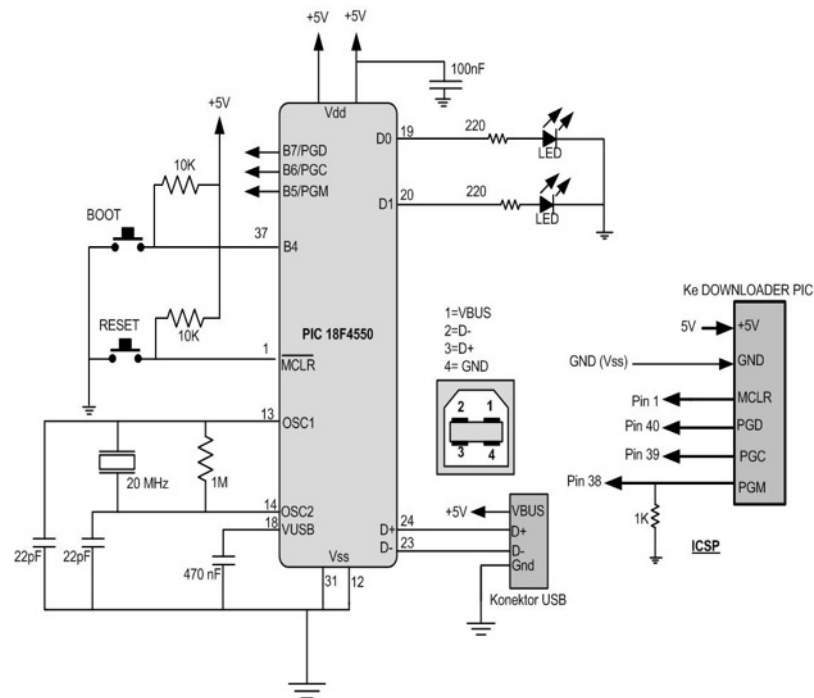
Gambar 3.6. Salah satu proses instalasi Microchip Application

- File source code bootloader dapat ditemukan di directory :
C:\Microchip Solutions\USB Device - Bootloaders\Vendor Class - MCHPUSB Bootloader\Bootloader - Firmware for PIC18F4550 Family Devices.
Buka File ***MCHPUSB.mcp*** pada directory tersebut dengan MPLAB IDE. ***MCHPUSB.mcp*** merupakan file project untuk bootloader PIC seri 18F dari Microchip, dengan standar bahasa Microchip C18. Oleh karena itu C18 juga harus sudah terinstall dengan benar, dan menjadi kompilator standar di MPLAB IDE.
- Setelah ***MCHPUSB.mcp*** terbuka pada MPLAB IDE, lakukan pengkompilan dengan mengklik **Project> Build All...** Jedela Output yang menunjukkan kompilasi berhasil akan muncul seperti Gambar 3.7.



Gambar 3.7. Proses kompilasi berhasil, file output (.hex) ada di folder yang sama dengan file project.

- Tutup MPLAB IDE. Dengan demikian kita sudah mempunyai file ***MCHPUSB.hex*** yang siap di bakar (burning) ke PIC18F4550. ***MCHPUSB.hex*** merupakan firmware bootload yang harus ditanam di PIC18F4550.
- Dengan menggunakan Downloader (WinPic800 atau PICPgm) upload ***MCHPUSB.hex*** (dalam directory ***C:\Microchip Solutions\USB Device - Bootloaders\Vendor Class - MCHPUSB Bootloader\Bootloader - Firmware for PIC18F4550 Family Devices***) ke PIC18F4550.
 - Untuk menjalankan aplikasi bootloader, kita harus membuat sistem minimal untuk PIC standard bootloader. Rangkaian dasar untuk standatd bootload ditunjukkan oleh Gambar 3.8.



Gambar 3.8 Rangkaian standard keperluan bootloader. LED yang terhubung ke Port D0 dan D1 merupakan penanda bahwa bootloader sudah berjalan atau belum.

MASUK MENU BOOTLOADER

Jika PIC18F4550 sudah terisi program MCHPUSB.hex dan sudah berada pada rangkaian seperti Gambar 3.8, maka kita siap memasuki menu bootloader PIC 18F4550. Langkah-langkahnya adalah sebagai berikut:

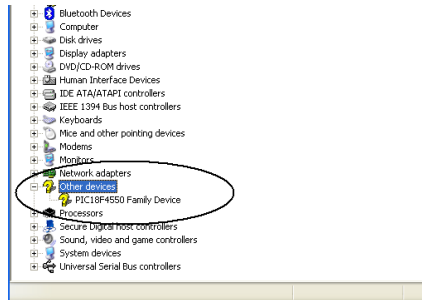
1. Untuk menggunakan bootloader agar mampu mendownload sebuah file .hex ke dalam PIC maka kita harus masuk ke dalam mode *Update Firmware* dengan cara:
 - a. Tekan Saklar *Boot* beberapa saat.
 - b. Sambil menekan saklar *Boot*, tekan saklar *Reset* sekali, sampai terlihat LED pada D0 dan D1 berkedip, lalu lepaskan saklar *Boot*.
 - c. Sekarang LED pada D0 dan D1 akan nyala flip flop bergantian. Artinya PIC sudah siap untuk masuk bootloader. Jika LED belum berkedip lakukan kembali langkah “a” sampai “c” di atas.
 - d. Indikasi bahwa kita sudah masuk ke menu bootloader adalah sesuai keadaan flip flop pada LED yang terhubung dengan Port D0 dan D1.

Sekarang, Windows akan menanyakan driver, apalagi kelas yang dipakai adalah custom/kelas khusus. Install driver untuk Windows dengan memilih file sendiri sebagai driver yaitu di directory:

C:\Microchip Solutions\USB Tools\MCHPUSB Custom Driver\MCHPUSB Driver\Release

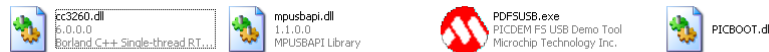
pilih file **mchpusb.inf**.

Pemilihan driver tersebut akan menyebabkan *device manager* pada Ms Windows akan mengidentifikasi device yang masuk sebagai keluarga PIC18F4550 seperti terlihat pada Gambar 3.9.

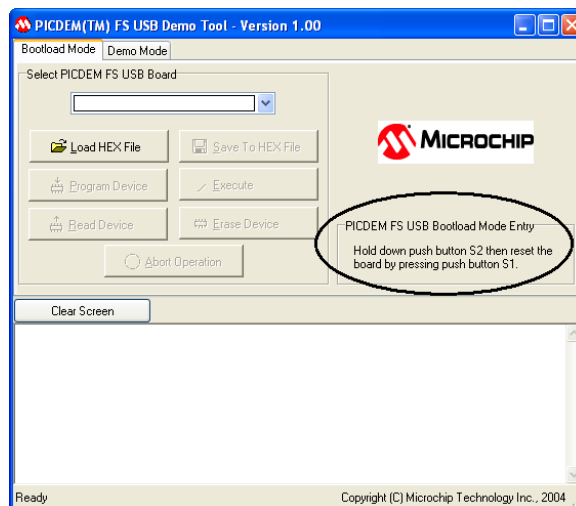


Gambar 3.9. Identifikasi Kelas Khusus USB pada Device Manager, ketika PIC masuk Mode Update Firmware

2. Beralih ke sisi PC atau MS Windows. Buka *C:\Microchip Solutions\USB Tools\Pdfusb*, lalu buka *PDFSUSB.exe* (Gambar 3.10). Aplikasi ini merupakan tool dari Microchip untuk menjalankan bootload dari sisi PC/ host. Pertama kali aplikasi *PDFSUSB* dibuka akan tampil pilihan PICDEM FS USB Board seperti Gambar 3.11.



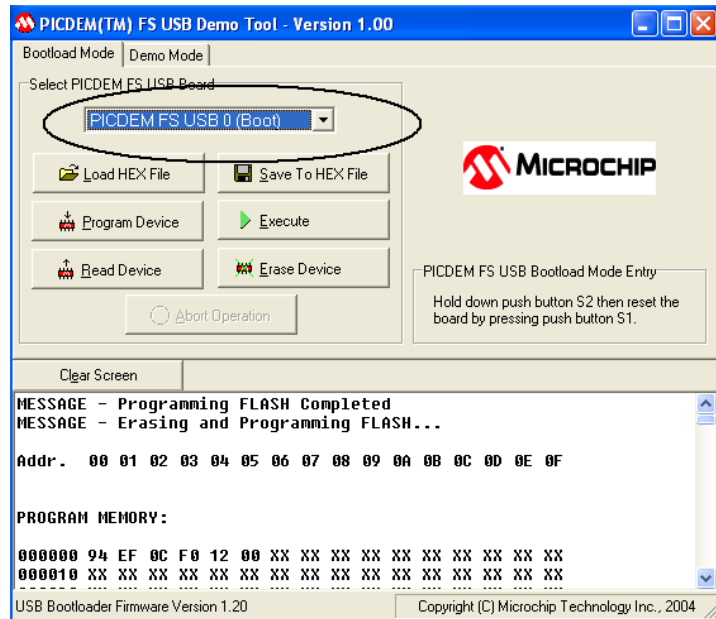
Gambar 3.10. Aplikasi Bootloder dari sisi host dengan PDFSUSB



Gambar 3.11. Aplikasi PICDEM FS USB Bootload pertama kali dijalankan.

Pada jendela PICDEM FS USB DEMO, sekarang klik pilihan Select PICDEM FS USB Board (Gambar 3.12) dan pilih PICDEM FS USB 0 (Boot). Apabila pilihan belum keluar dan masih kosong berarti aplikasi belum mendeteksi adanya bootload USB pada PIC18F4550 atau PIC belum masuk ke Mode Update Firmware atau dari sisi Windows ada kesalahan dalam mendeteksi kelas khusus PIC. Jika hal itu terjadi, ulangi langkah masuk ke *update firmware* atau menu bootload dari awal.

Setelah semua langkah terlampaui (tampilan di Windows sudah seperti Gambar 3.12), berarti program bootload telah siap digunakan untuk meload program .HEX, tanpa rangkaian downloader lagi, sehingga kita bisa mengupdate, dan mengganti firmware kita sewaktu-waktu dengan mudah. Tetapi program .HEX yang berjalan di atas bootload mempunyai perbedaan dengan *.HEX yang berjalan di atas memori biasa. Perbedaannya adalah setting memori data yang di set di atas lokasi memori bootload.



Gambar 3.12. PICDEM FS USB Bootload setelah mendeteksi PIC18F4550 masuk ke mode Update Firmware

Sekali kita masuk ke dalam mode bootload, maka kita bisa mengupload kode firmware kita, setelah selesai, tekan tombol Reset kembali pada sistem minimum bootload PIC, maka PIC kembali ke mode user, dan menjalankan firmware yang baru. Program bootload akan selalu ada sepanjang kita tidak merusak memori data yang digunakan oleh bootload. Pada kesempatan kedua dan seterusnya proses update firmware bisa dilakukan melalui bootloader. Langkah-langkah masuk ke bootload juga sama yaitu dengan menekan tombol Boot beberapa saat diikuti dengan Reset. Setelah itu kita tinggal memilih firmware yang baru.