

BAB II

TINJAUAN PUSTAKA

2.1 Arduino Software (IDE)

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.



Gambar 2.1 Arduino Software
(Sumber: Data penulis)

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software *Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Program yang ditulis dengan menggunakan Arduino Software (IDE) disebut sebagai *sketch*. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi.

Teks editor pada Arduino Software ini memiliki fitur seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan dalam menulis kode program. Pada Software Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta *COM Ports* yang digunakan. Jika sebuah sketch sedang berjalan pada board menerima satu kali konfigurasi atau data lain ketika *sketch* pertama mulai, memastikan bahwa *software* yang berkomunikasi menunggu satu detik setelah membuka koneksi dan sebelum mengirim data ini. Untuk dapat melakukan pemrograman dengan benar maka Arduino IDE harus dikoneksikan dengan board Arduino yang telah terinstall pada *port* tertentu. Untuk mengupload program kemikrokontroller dapat menggunakan kabel USB sebagai medianya.



Gambar 2.2 Layar utama Arduino Software

(Sumber: Data penulis)

2.2 Instalasi Software IDE ARDUINO

Software IDE Arduino dapat diunduh pada www.arduino.cc/en/Main/Donate, kemudian akan tampil dua pilihan, pilih download and donate dan just download. Unduh file arduino-1-6-3-windows.exe setelah selesai diunduh klik dua kali file arduino file arduino-1-6-3-windows.exe lalu akan tampil seperti gambar berikut ini.



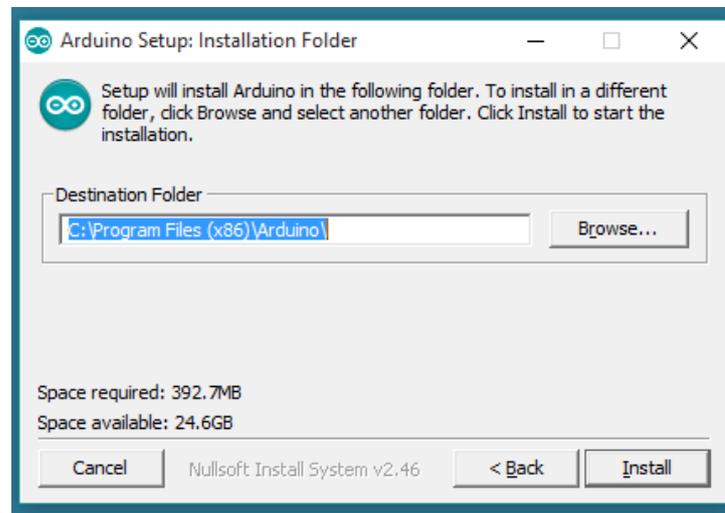
Gambar 2.3 Persetujuan instalasi arduino software
(Sumber: Data Penulis)

Klik tombol I Agree, lalu akan muncul Installation Option, pilih semuanya, termasuk install USB driver untuk mengenali dan melakukan komunikasi dengan board arduino melalui port USB.



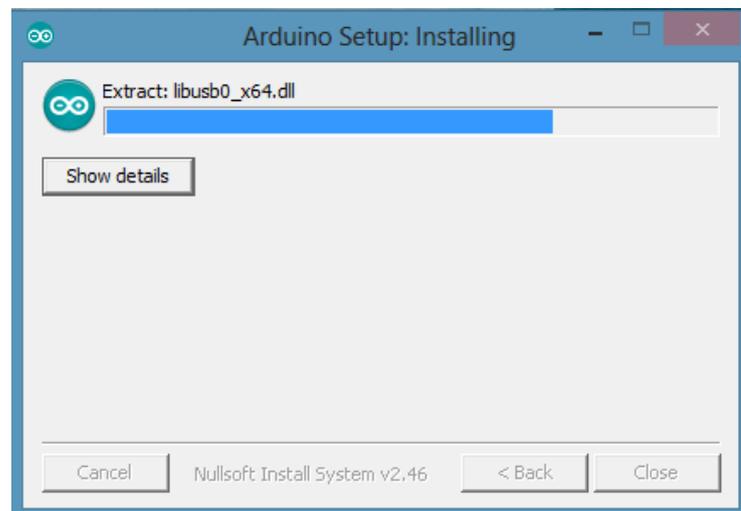
Gambar 2.4 Pilihan instalasi
(Sumber: Data Penulis)

Klik next, lalu pilih folder untuk menyimpan program arduino.



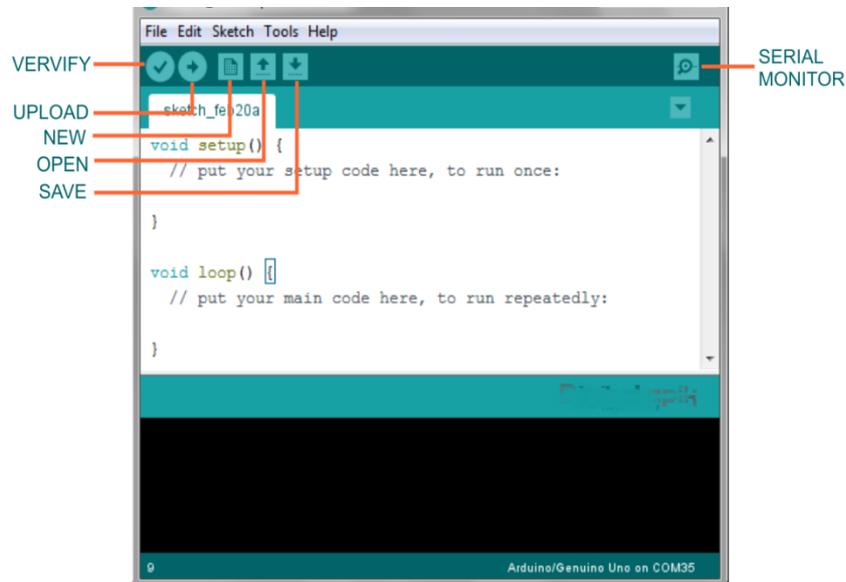
Gambar 2.5 Instalasi Folder
(Sumber: Data Penulis)

Klik Install, untuk melakukan proses instalasi, tunggu sampai Finish, lalu klik.



Gambar 2.6 Proses extract
(Sumber: Data Penulis)

2.3 Pengenalan Software Arduino IDE



Gambar 2.7 Tampilan Awal Arduino IDE
(Sumber: Data Penulis)

Arduino Integrated Development Environment - atau Arduino Software (IDE) - berisi editor teks untuk menulis kode, area pesan, konsol teks, toolbar dengan tombol untuk fungsi-fungsi umum dan serangkaian menu. Termasuk menghubungkan ke perangkat keras Arduino untuk meng-upload program dari komputer.

- ✓ *Verify*: Memeriksa kode Anda untuk kesalahan kompilasi.
- ➔ *Upload* : Mengkompilasi kode Anda dan mengunggahnya ke papan yang dikonfigurasi. Lihat pengunggahan di bawah untuk detail. Catatan: Jika Anda menggunakan pemrogram eksternal dengan papan Anda, Anda dapat menahan tombol “Shift” pada keyboard ketika menggunakan ikon ini. Teks akan berganti menjadi “Upload using Programmer” (Unggah menggunakan Programmer).

📄 *New*: Membuat sketsa baru

 *Open*: Menampilkan sebuah menu dari semua sketsa pada *sketchbook* (buku sketsa) Anda. Mengklik salah satu akan membukanya ke dalam jendela saat ini dan menggantinya isinya.

 *Save*: Menyimpan sketsa Anda.

 *Serial Monitor*: Membuka serial monitor.

Perintah tambahan dapat ditemukan dalam lima menu: **File**, **Edit**, **Sketch**, **Tool**, **Help**. Menu-menu ini *context sensitive*, artinya *item-item* yang tersedia hanya yang relevan dengan pekerjaan yang bersangkutan.

File

1. *New*

Membuat sebuah sketsa baru pada editor, termasuk dengan struktur minimal sketsa.

2. *Open*

Memungkinkan membuka sketsa yang ada pada media penyimpanan komputer.

3. *Open Recent*

Menyediakan daftar pendek sketsa yang baru-baru yang siap dibuka.

4. *Sketchbook*

Menampilkan sketsa saat ini dalam struktur folder *sketchbook*; mengklik salah satu nama akan membuka sketsa yang bersangkutan pada editor baru.

5. *Examples*

Beberapa contoh telah disediakan oleh Arduino Software (IDE) atau pustaka yang ditampilkan dalam menu *item* ini. Semua contoh tersusun dalam struktur pohon yang mudah diakses.

6. *Close*

Menutup jendela Arduino Software yang bersangkutan.

7. *Save*

Menyimpan sketsa dengan nama yang bersangkutan, jika belum dinamai sebelumnya, bisa dinamai pada jendela “Save as...”.

8. *Save as*

Memungkinkan menyimpan sketsa yang bersangkutan dengan nama yang berbeda.

9. *Page Setup*

Menampilkan jendela “Page Setup” (pengaturan halaman) untuk pencetakan.

10. *Print*

Mengirim sketsa saat ini ke printer berdasarkan pengaturan yang ditetapkan pada Page Setup.

11. *Preferences*

Membuka jendela Preferences dimana beberapa pengaturan IDE bisa dikustomisasi.

12. *Quit*

Menutup semua jendela IDE. Sketsa yang terbuka ketika Quit dipilih akan otomatis dibuka kembali pada waktu memulai IDE yang selanjutnya.

Edit

1. *Undo/Redo*

Kembali satu atau lebih tahap yang Anda lakukan saat editing; ketika Anda memilih Undo, Anda dapat maju satu atau lebih tahap kembali dengan Redo.

2. *Cut*

Memindahkan teks yang diseleksi dari editor ke papan klip (*clipboard*).

3. *Copy*

Menyalin teks yang diseleksi dari editor ke papan klip.

4. *Copy For Forum*

Menyalin kode dari sketsa Anda ke papan klip dalam suatu form yang cocok diposting ke forum, lengkap dengan pewarnaan sintaks.

5. *Copy as HTML*

Menyalin kode dari sketsa Anda ke papan klip sebagai HTML, cocok di *embed* (tanam) pada laman web

6. *Paste*

Meletakkan isi papan klip tepat pada posisi kursor dalam editor.

7. *Select All*

Menyeleksi dan menyorot seluruh isi editor.

8. *Comment/Uncomment*

Memberi atau menghapus penanda komen “//” pada awal baris yang diseleksi.

9. *Increase/Decrease Indent*

Menambah atau mengurangi sebuah spasi pada awal setiap baris yang diseleksi, memindahkan teks satu spasi ke kanan atau menghapus satu spasi di awal.

10. *Find*

Membuka jendela “Find and Replace” dimana Anda dapat mengetik teks untuk dicari dalam sketsa yang bersangkutan berdasarkan beberapa pilihan.

11. *Find Next*

Menyorot kejadian selanjutnya (jika ada) dari kata yang diketik pada jendela “Find”, relatif ke posisi kursor.

12. *Find Previous*

Menyorot kejadian sebelumnya (jika ada) dari kata yang diketik pada jendela “Find”, relatif ke posisi kursor.

Sketch

1. *Verify/Compile*

Memeriksa sketsa Anda untuk kesalahan kompilasi, ini akan melaporkan penggunaan memori kode dan variabel pada konsol.

2. *Upload*

Mengkompilasi dan memuat berkas *binary* ke papan yang dikonfigurasi melalui Port yang dikonfigurasi.

3. *Upload Using Programme*

Ini akan menulis ulang *bootloader* pada papan, Anda akan perlu menggunakan *Tools > Burn Bootloader* untuk memperbaikinya dan memungkinkan Mengunggah ke port USB serial kembali. Bagaimanapun, ini memungkinkan Anda untuk menggunakan kapasitas penuh memori Flash untuk sketsa Anda. Tolong catat bahwa perintah ini TIDAK akan membakar sekering. Untuk melakukannya perintah *Tools > Burn Bootloader* harus dieksekusi.

4. *Export Compiled Binary*

Menyimpan sebuah berkas *.hex* yang bisa diarsipkan atau dikirim ke papan menggunakan alat lain.

5. *Show Sketch Folder*

Membuka folder sketsa yang bersangkutan.

6. *Include Library*

Menambah pustaka ke sketsa Anda dengan memasukkan *statements #include* pada awal kode Anda. Untuk detail lihat pustaka di bawah. Anda juga dapat mengakses *Library Manager* dan mengimpor pustaka baru dari berkas *.zip*.

7. *Add File*

Menambah sebuah berkas sumber ke sketsa (disalin dari lokasinya). Berkas baru akan muncul dalam tab baru pada jendela sketsa. Berkas dapat dihapus dari sketsa

menggunakan menu tab (ikon segitiga kecil dibawah ikon serial monitor pada sisikanan *toolbar*).

Tools

1. *Auto Format*

Memformat kode Anda dengan baik. Contoh memasukkannya jadi kurung kurawal membuka dan menutup lurus, dan teks dalam kurung kurawal juga diluruskan.

2. *Archive Sketch*

Mengarsipkan sebuah salinan dari kode yang bersangkutan ke format .zip. Arsip ini ditempatkan pada direktori yang sama dengan sketsa.

3. *Fix Encoding & Reload*

Memperbaiki ketidaksesuaian yang mungkin antara peta karakter *encoding* dan peta karakter sistem operasi lain.

4. *Serial Monito*

Membukan jendela *serial monitor* dan menginisialisasi perubahan data dengan suatu papan yang terhubung pada port yang bersangkutan. Biasanya mengatur ulang papan, jika papan itu mendukung pengaturan ulang diatas pembukaan port serial.

5. *Board*

Memlilih papan yang Anda gunakan. Lihat di bawah untuk deskripsi berbagai macam papan.

6. *Port*

Menu ini mengandung semua perangkat serial (nyata atau virtual) pada mesin Anda. Seharusnya otomatis *me-refresh* setiap saat Anda membuka menu *tools* tingkat atas.

7. *Programmer*

Untuk menyeleksi sebuah perangkat keras pemrogram ketika memrogram sebuah papan atau *chip* dan tidak menggunakan hubungan USB-serial *onboard*. Secara norma Anda tidak membutuhkan ini, tetapi jika Anda membakar *bootloader* ke mikrokontroler baru, Anda akan menggunakan ini.

8. *Burn Bootloader*

Memungkinkan Anda membakar sebuah *bootloader* ke mikrokontroler pada papan Arduino. Ini tidak dibutuhkan untuk penggunaan normal papan Arduino atau Genuino, tetapi sangat berguna jika Anda membeli mikrokontroler Atmega baru (secara normal tanpa *bootloader*). Pastikan Anda telah memilih papan yang benar pada menu **Boards** sebelum membakar *bootloader* ke papan target. Perintah ini juga mengatur sekering yang benar.

Help

Disini Anda memnemukan kemudahan akses ke sejumlah dokumen yang datang bersama Arduino Software (IDE). Anda bisa mengakses ke *Getting Started*, *Referensi*, petunjuk IDE ini dan dokumen lain secara lokal tanpa koneksi internet.

1. *Find in Reference*

Fungsi interaktif pada menu **Help**: secara langsung memilih laman yang relevan dalam salinan lokal dari Referensi untuk fungsi atau perintah pada kursor.

2.4 Pemrograman Arduino

Pemrograman arduino menggunakan struktur Bahasa C. Mekanisme pemrogramanya arduino sama dengan mikrokontroler pada umumnya. Mulai dari membuat sket progam, meng-*compile*, selanjutnya proses *upload* pada papan arduino. Pengisian progam dengan metode upload ialah mengisi papan arduino dengan progam yang sudah berbentuk Hex atau hasil *compile* dari bahasa C ke bahasa mesin.



Gambar 2.8 Mekanisme Pemograman

2.4.1 Struktur Utama

1. Setup ()

Fungsi setup() dipanggil ketika sketsa program dimulai. Fungsi ini digunakan untuk menginisialisasi variabel, mode pin, penggunaan librari, dll. Fungsi setup() hanya akan berjalan sekali, setelah power arduino dinyalakan atau saat mereset papan Arduino.

Contoh : Program 1.1

```

int ledPin = 13;
void setup(){
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);
  delay(5000);
  digitalWrite(ledPin, HIGH);
}
void loop(){
  // ...
}
  
```

Program 1.1 akan menyalakan LED pada pin 13 selama 5 detik lalu mati. Eksekusi ini dilakukan hanya sekali.

2. Loop ()

Berikutnya adalah fungsi loop().

```
const int buttonPin = 3;
setup ()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}
loop()
{
  if (digitalRead (buttonPin) == HIGH)
    Serial.write ( 'H');
  else
    Serial.write ( 'L');
  delay (1000);
}
```

Program 1.2 pada serial monitor akan menampilkan huruf H ketika tombol pada pin 3 ditekan dan bila dilepaskan akan tampil huruf L.

Fungsi loop() akan melakukan loop berturut-turut dimana program akan dijalankan terus menerus secara beruruta dan loop untuk mengontrol papan Arduino.

3. //Komentar

Komentar digunakan untuk memberikan keterangan pada program yang dibuat. Komentar tidak dieksekusi maka komentar tidak menambah ukuran file hasil *compile*. Cara membuat komentar ialah sebagai berikut :

```
//komentar segaris diawali dengan dua garis miring
/*komentar untuk lebih dari satu baris diawali
dengan garis miring lalu tanda bintang serta
diakhiri dengan bintang lalu garis miring*/
```

Elemen bahasa C yang dibutuhkan untuk format penulisan.

1. // (komentar satu baris)

Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

2. /* */ (komentar banyak baris)

Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

3. { } (kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

4. ; (titik koma)

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

2.4.2 Struktur Kontrol

Setiap program yang dibuat membutuhkan suatu kontrol. Tak hanya perulangan namun suatu eksekusi dengan syarat tertentu juga diperlukan.

Pengujian Kondisi :

- **if**

Digunakan untuk mengecek suatu kondisi. Jika benar maka perintah didalam **if** akan dikerjakan.

```
if(kondisi){
    Pernyataan / perintah
}
```

Contoh :

```
if(x==6)
{
    a=a+
    5;
}
```

- **if – else**

seperti dengan **if**, hanya saja ada 2 pilihan pernyataan / perintah. Jika

kondisi benar maka perintah didalam **if** akan dikerjakan, jika kondisinya salah maka pernyataan didalam **else** yang akan dikerjakan

```

if(kondisi){
    Pernyataan / perintah 1
}
else {
    Pernyataan / perintah 2
}

```

Contoh :

```

if(x==1)
    { a=1;
    }
else {
    a=0;
    }

```

- **if – else if**

Untuk melakukan pengecekan suatu kondisi lebih dari satu maka bisa menggunakan **if – else if**.

```

if(kondisi1){
    Pernyataan / perintah 1
}
else if(kondisi2){ Pernyataan / perintah 2
}
else if(kondisi ke-n){ Pernyataan / perintah ke-n
}

```

Contoh :

```

if(x==1){

```

```

        a=1;
    }
    else if(x==2){ a=2;
    }
    else if(x==3){
        a=3;
    }

```

2.4.3 Switch Case

Pernyataan ini digunakan untuk memilih kondisi yang sesuai untuk kemudian mengerjakan perintahnya. Bedanya adalah kondisi yang diuji berupa sebuah nilai variable.

```

switch(variabel){ //variable yang diuji
    case 1    ://pernyataan/perintah 1 break;
    case 2    ://pernyataan/perintah 2
                break;
    case n    ://pernyataan/perintah n
                break;
    default  : //pernyataan/perintah default
}

```

Jika variable memenuhi syarat dari salah satu case maka dia akan mengerjakan pernyataan/perintah tersebut. Misal nilai variable = 2 maka dia akan mengerjakan pernyataan/perintah 2. Jika tidak memenuhi maka dia akan mengerjakan default.

Contoh :

```

switch(a){
    case 1    :    digitalWrite(pin1,HIGH)
                break;

```

```
case 2    :    digitalWrite(pin2,HIGH)
break;
case 3    :    digitalWrite(pin3,HIGH)
break;
default  : digitalWrite(pin4,LOW)
```

2.5 Internet Of Thing (IoT)

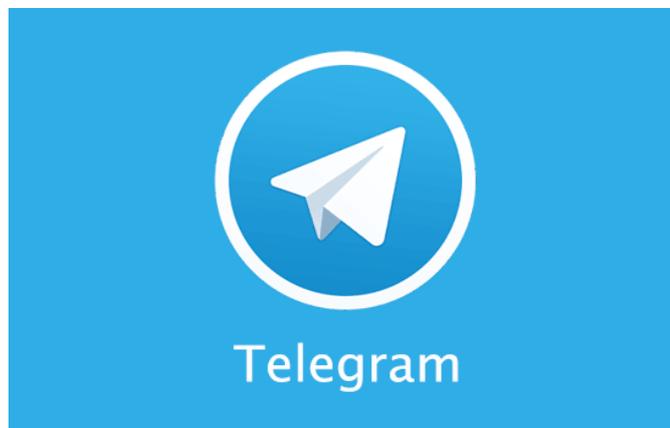
Jadi *Internet of Thing* (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, *micro electromechanical systems* (MEMS), dan Internet.

“A Things” pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan transponder biochip, sebuah mobil yang telah dilengkapi built-in sensor untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, IoT paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau “*smart*”. Sebagai contoh yaitu smart kabel, smart meter, smart grid sensor.

2.6 Aplikasi Telegram

Telegram merupakan aplikasi *chatting* yang ringan, cepat, tidak ada iklan dan gratis. Aplikasi ini memiliki sistem bot atau biasa disebut telegram bot yang dapat digunakan untuk komunikasi dengan perangkat mikrokontroler. Telegram adalah layanan pesan instan berbasis *cloud* dan gratis. Klien Telegram ada untuk selulaer dan dekstop. Pengguna dapat mengirim pesan dan bertukar foto, video, stiker, audio, dan file jenis apapun.

Telegram adalah sebuah aplikasi yang memungkinkan pengguna atau User untuk mengirimkan pesan *Chatting* rahasia atau *Secret Chat* yang di enkripsi *end-to-end* sebagai keamanan tambahan. Dengan menggunakan Telegram anda juga bisa mengirim bukan hanya sekedar gambar dan video, tapi anda juga bisa mengirim dokumen seperti *word*, *excel*, PDF dan lainnya tanpa menetapkan besarnya *size file* yang di kirimkan, juga bisa mengirimkan lokasi anda dengan mudah Lalu anda bisa menggunakan Telegram dengan menggunakan PC atau komputer.



Gambar 2.9 Aplikasi Telegram
(Sumber: Ilustrasi Telegram. Lifewire.com)

2.6.1. Cara penginstalan Telegram

Berikut adalah cara *download* dan install aplikasi Telegram di komputer :

1. Untuk mengunduh aplikasi Telegram silahkan menuju ke situs download Telegram. <https://desktop.telegram.org/>
2. Klik tombol Get Telegram for Windows. Lanjutkan sampai proses unduhan selesai. (di sini sistem digunakan Windows) adalah



Telegram Desktop

Fast and secure desktop app, perfectly synced with your mobile phone.

Get Telegram for **Windows**

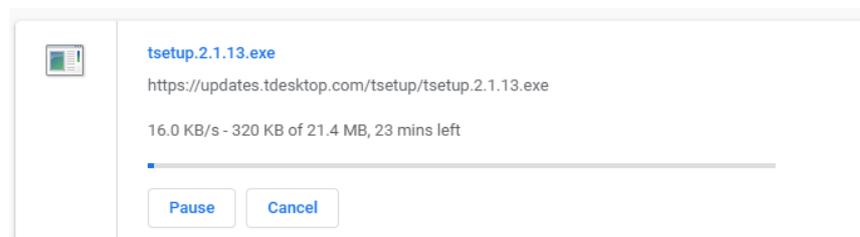
Portable version for Windows

Show all platforms

Gambar 2.10 Tampilan instal Telegram

(Sumber: Data Penulis)

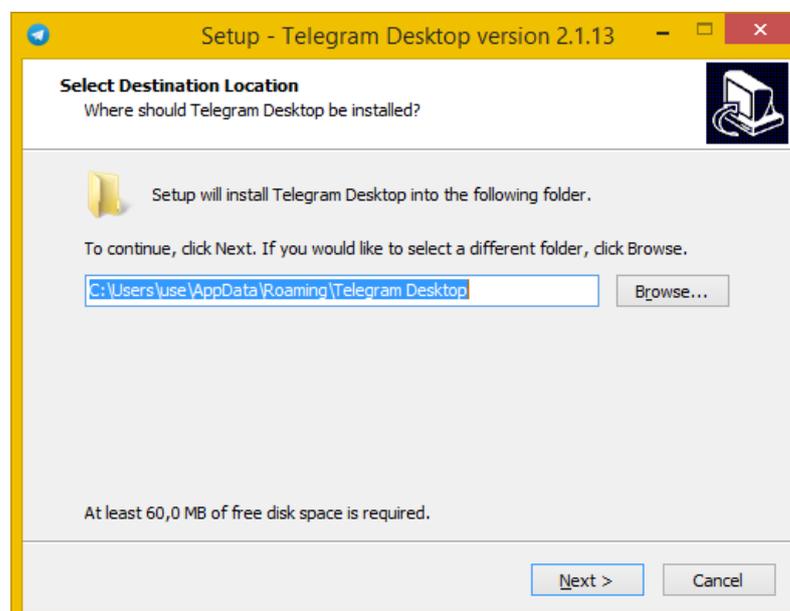
3. Tunggu proses unduhan.



Gambar 2.11 Tampilan proses penginstalan

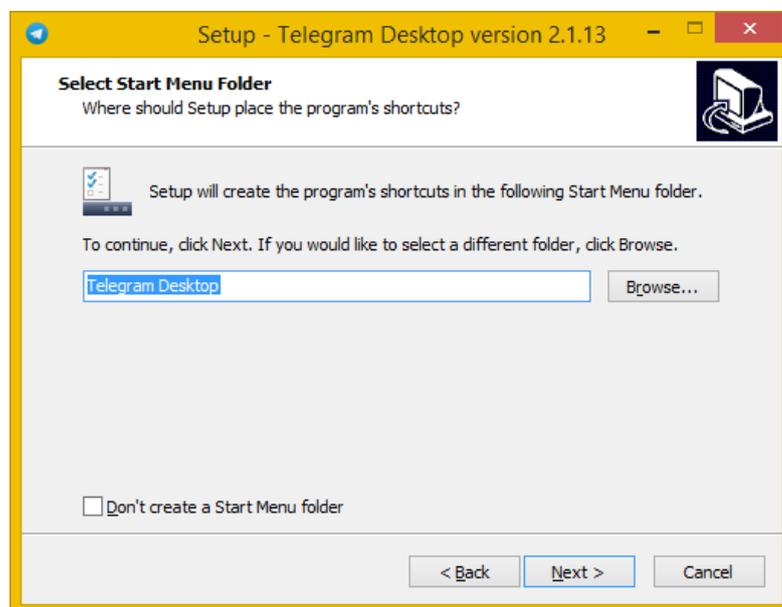
(Sumber: Data Penulis)

4. Setelah selesai terunduh, install aplikasi Telegram dengan cara membuka file hasil unduh tersebut.
5. Muncul opsi tentang lokasi file. Klik tombol Next.



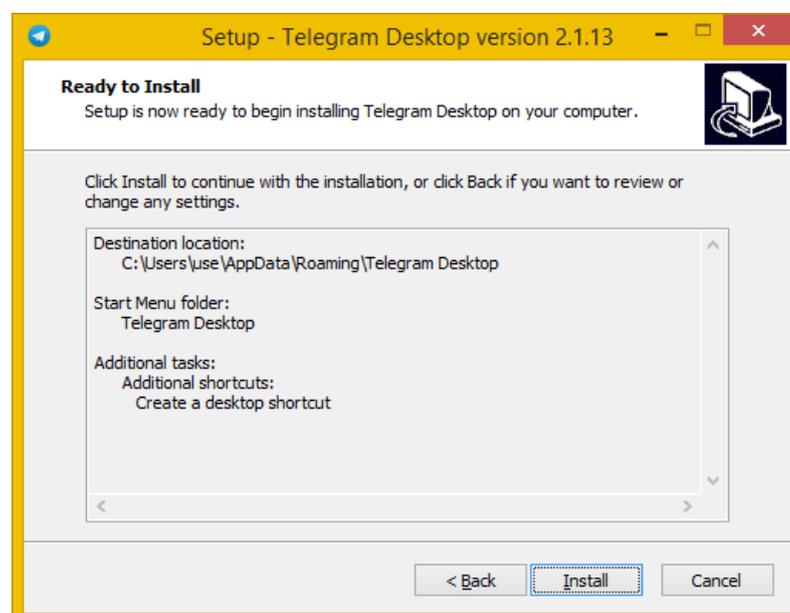
Gambar 2.12 Tampilan letak file
(Sumber: Data Penulis)

6. Muncul opsi tentang shortcut di Desktop. Klik tombol Next.



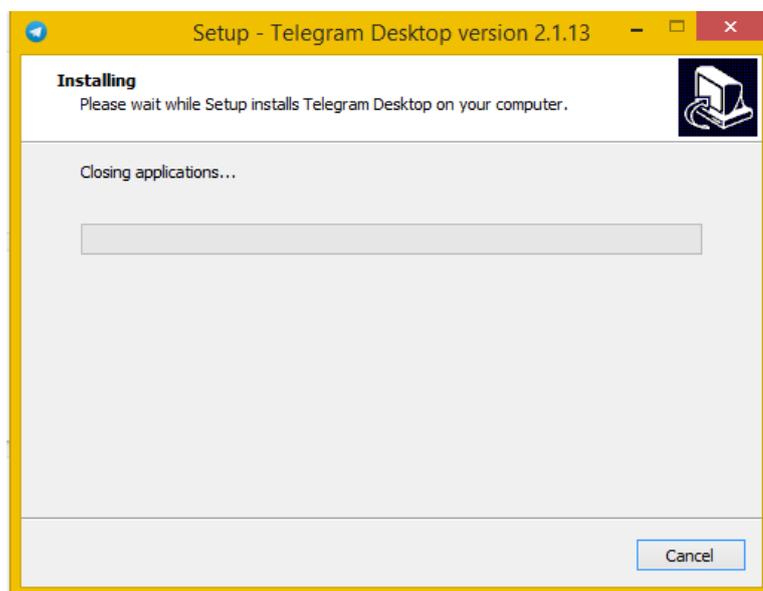
Gambar 2.13 Tampilan menu folder
(Sumber: Data Penulis)

7. Klik tombol Install.



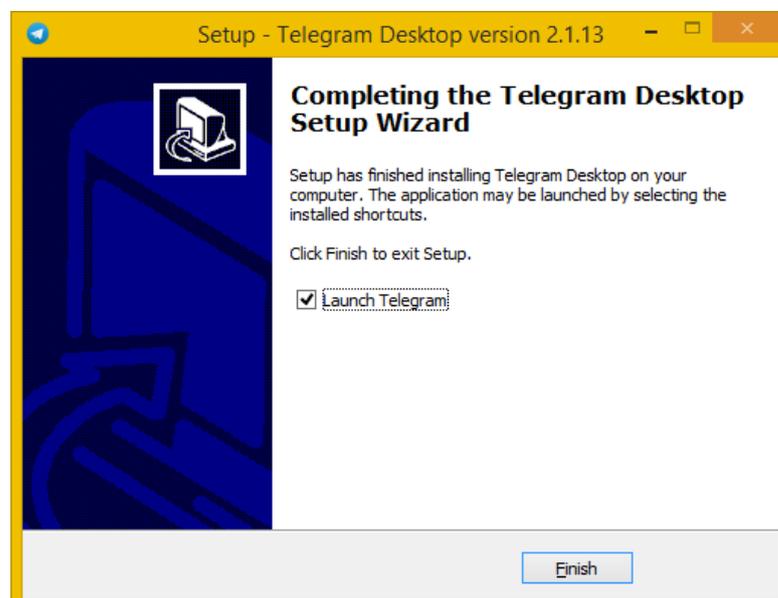
Gambar 2.14 Tampilan ready to install
(Sumber: Data Penulis)

8. Tunggu proses install.



Gambar 2.15 Tampilan proses penginstalan
(Sumber: Data Penulis)

9. Setelah proses install berhasil, klik tombol Finish.



Gambar 2.16 Instal Finish
(Sumber: Data Penulis)