



TUTORIAL BOT TELEGRAM MATA KULIAH DIGITAL KREATIF
PROGRAM STUDI SISTEM INFORMASI KAMPUS UTAMA DAN PSDKU
FAKULTAS TEKNIK DAN INFORMATIKA

PERANGKAT LUNAK YANG HARUS DIINSTAL DI KOMPUTER LOKAL

1. Download dan Install node js dan npm

<https://nodejs.org/en/download/>

pastikan node js versi 22

Download Node.js®

The screenshot shows the official Node.js download page. At the top, there are dropdown menus for selecting the version (v22.20.0 LTS), operating system (Windows), and npm version (npm 8). Below these, a green bar displays the message: "Info Want new features sooner? Get the [latest Node.js version](#) instead and try the latest improvements!" A code block shows Docker installation steps:

```
1 # Docker has specific installation instructions for each operating system.
2 # Please refer to the official documentation at https://docker.com/get-started/
3
4 # Pull the Node.js Docker image:
5 docker pull node:22-alpine
6
7 # Create a Node.js container and start a Shell session:
8 docker run -it --rm --entrypoint sh node:22-alpine
9
10 # Verify the Node.js version:
11 node -v # Should print "v22.20.0".
12
13 # Verify npm version:
14 npm -v # Should print "10.9.3".
```

Below the code block, it says "PowerShell" and "Copy to clipboard". Further down, it says "Docker is a containerization platform. If you encounter any issues please visit [Docker's website](#)". At the bottom, it says "Or get a prebuilt Node.js® for Windows running a x64 architecture." with two download buttons: "Windows Installer (.msi)" and "Standalone Binary (.zip)".

link tutorial : <https://www.geeksforgeeks.org/node-js/how-to-download-and-install-node-js-and-npm/>

Setelah instalasi selesai, verifikasi bahwa **Node.js** dan **NPM** telah terpasang dengan benar dengan menjalankan langkah-langkah berikut:

A screenshot of a Windows Command Prompt window titled "cmd.exe C:\Windows\system32\cmd.exe". The window shows the following text:

```
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>node -v
v22.20.0

C:\Users\User>npm -v
10.9.3
```

Perintah di atas digunakan untuk mengecek versinya. Jika hasilnya muncul seperti pada gambar, berarti instalasi berhasil

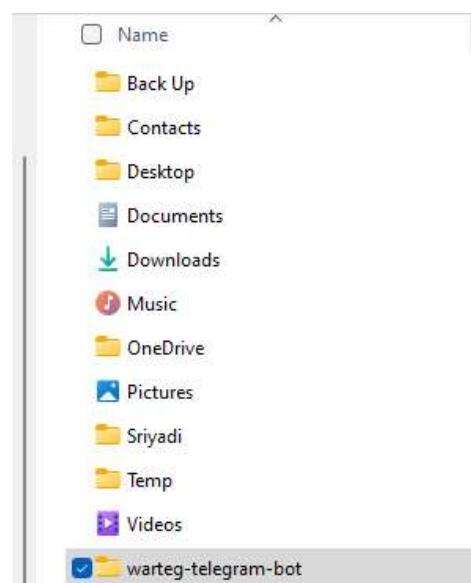
2. Menyiapkan proyek bot Telegram



Proyek bot akan disimpan di Local Disk (D) dengan membuat folder bernama warteg-telegram-bot. Ikuti langkah-langkah sesuai dengan gambar yang ditampilkan.

```
C:\Users\User>d:  
D:\>mkdir warteg-telegram-bot  
D:\>cd warteg-telegram-bot
```

Buka Windows Explorer dan pastikan folder warteg-telegram-bot sudah terlihat di Drive D



3. Instal library Telegram di Node.js. Pastikan terminal yang aktif berada di dalam folder warteg-telegram-bot, seperti yang ditunjukkan pada gambar. Kemudian jalankan perintah berikut: **npm init -y**

```
D:\warteg-telegram-bot>npm init -y  
Wrote to D:\warteg-telegram-bot\package.json:  
  
{  
  "name": "warteg-telegram-bot",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\"Error: no test specified\\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```

Lanjutkan dengan menjalankan perintah berikut untuk menginstal library: **npm install node-telegram-bot-api** Pastikan hasilnya sesuai dengan yang ditampilkan pada gambar.

```
D:\warterg-telegram-bot>npm install node-telegram-bot-api
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated uidash@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm warn deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
added 379 packages, and audited 188 packages in 10s

73 packages are looking for funding
  run 'npm fund' for details

0 vulnerabilities (0 moderate, 0 critical)

To address all issues, run:
  npm audit fix

Run 'npm audit' for details.
```

Pada gambar di atas, terlihat perintah **npm audit fix**. Jalankan perintah tersebut di terminal untuk memperbaiki potensi kerentanan yang terdeteksi. Hasilnya seperti pada gambar dibawah

```
D:\warterg-telegram-bot>npm audit fix
up to date, audited 180 packages in 2s

73 packages are looking for funding
  run 'npm fund' for details

# npm audit report

form-data <2.5.4
Severity: critical
form-data uses unsafe random function in Form-data for choosing boundary - https://github.com/advisories/GHSA-fjxv-7rqg-78g4
Fix available via "npm audit fix --force"
Will install node-telegram-bot-api@0.63.0, which is a breaking change
node_modules/request/node_modules/form-data
  request *
    Depends on vulnerable versions of form-data
    Depends on vulnerable versions of tough-cookie
    node_modules/request
      request-promise-core *
        Depends on vulnerable versions of request
        node_modules/request-promise-core
          cypress/request-promise *
            Depends on vulnerable versions of request-promise-core
            node_modules/cypress/request-promise
              node-telegram-bot-api >=0.64.0
                Depends on vulnerable versions of @cypress/request-promise
                node_modules/node-telegram-bot-api

tough-cookie <4.1.3
Severity: moderate
tough-cookie Prototype Pollution vulnerability - https://github.com/advisories/GHSA-72xf-g2v4-qvf3
Fix available via "npm audit fix --force"
Will install node-telegram-bot-api@0.63.0, which is a breaking change
node_modules/request/node_modules/tough-cookie
  node_modules/request
    request *
      Depends on vulnerable versions of form-data
      node_modules/request-promise
        cypress/request-promise *
          Depends on vulnerable versions of request-promise-core
          node_modules/cypress/request-promise
            node-telegram-bot-api >=0.64.0
              Depends on vulnerable versions of @cypress/request-promise
              node_modules/node-telegram-bot-api

  # vulnerabilities (4 moderate, 2 critical)

  To address all issues (including breaking changes), run:
    npm audit fix --force
```

Perhatikan terdapat perintah : **npm audit fix –force**. berikan perintah tersebut pada prompt. Hasilnya seperti gambar berikut:

```
D:\warterg-telegram-bot>npm audit fix --force
npm WARN optional depening react-native@0.64.0 will be a hoisted major change
npm WARN optional depening react-native@0.64.0 react-native-promise has been deprecated because it contains the now deprecated request package, see https://github.com/react-native-community/realm#0.10.0
npm WARN optional depening react-native@0.64.0 react-native-promise has been deprecated because it contains the now deprecated request package, see https://github.com/react-native-community/realm#0.10.0
  added 1 package, removed 13 packages, changed 4 packages, and audited 187 packages in 2s

  73 packages are looking for funding
    run 'npm fund' for details

# npm audit report

form-data <2.5.4
Severity: critical
form-data uses unsafe random function in Form-data for choosing boundary - https://github.com/advisories/GHSA-fjxv-7rqg-78g4
Fix available via "npm audit fix --force"
Will install node-telegram-bot-api@0.63.0, which is a breaking change
node_modules/request/node_modules/form-data
  request *
    Depends on vulnerable versions of form-data
    Depends on vulnerable versions of tough-cookie
    node_modules/request
      request-promise-core *
        Depends on vulnerable versions of request
        node_modules/request-promise-core
          cypress/request-promise *
            Depends on vulnerable versions of request-promise-core
            node_modules/cypress/request-promise
              node-telegram-bot-api >=0.64.0
                Depends on vulnerable versions of @cypress/request-promise
                node_modules/node-telegram-bot-api

tough-cookie <4.1.3
Severity: moderate
tough-cookie Prototype Pollution vulnerability - https://github.com/advisories/GHSA-72xf-g2v4-qvf3
Fix available via "npm audit fix --force"
Will install node-telegram-bot-api@0.63.0, which is a breaking change
node_modules/request/node_modules/tough-cookie
  node_modules/request
    request *
      Depends on vulnerable versions of form-data
      node_modules/request-promise
        cypress/request-promise *
          Depends on vulnerable versions of request-promise-core
          node_modules/cypress/request-promise
            node-telegram-bot-api >=0.64.0
              Depends on vulnerable versions of @cypress/request-promise
              node_modules/node-telegram-bot-api

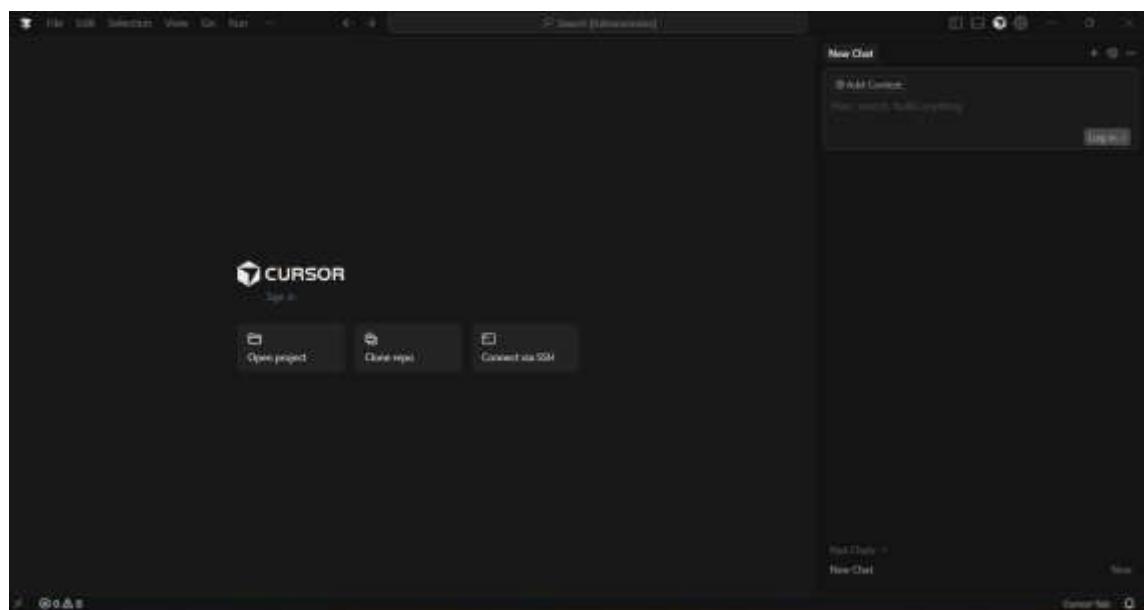
  # vulnerabilities (4 moderate, 2 critical)

  To address all issues (including breaking changes), run:
    npm audit fix --force
```

Proses selesai!

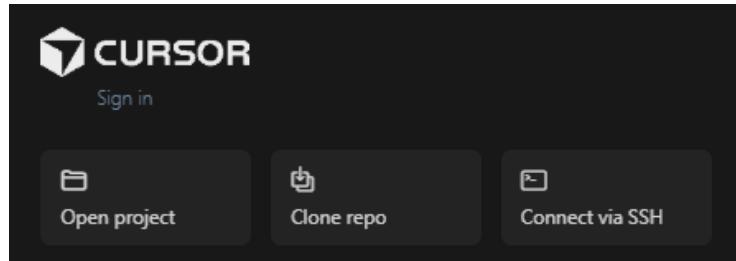
4. Install Cursor AI

<https://cursor.com/>

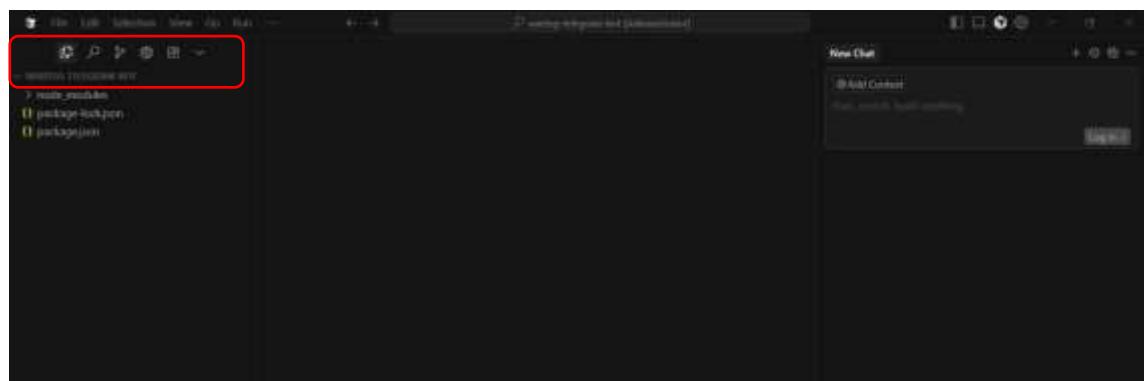


Tampilan Cursor AI setelah diinstal dan dijalankan. Dengan menggunakan Cursor AI, jika ditemukan bug, program ini dapat membantu memberikan solusi untuk bug tersebut.

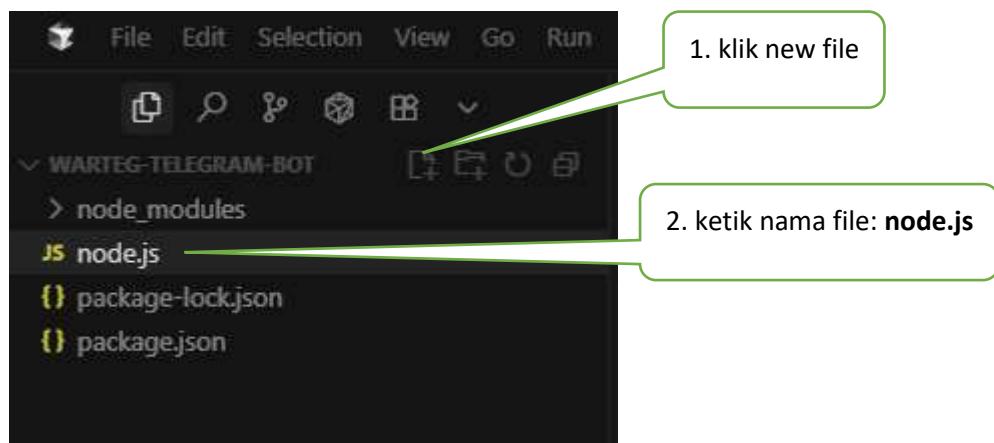
5. Menggunakan Cursor AI dalam proses kerja (coding and repair bug) aplikasi **bot telegram**
 - Open project



- Pastikan folder yang dibuka sebagai project adalah: **warteg-telegram-bot**



- Buat file baru dengan nama **bot.js** Perhatikan dengan seksama gambar berikut:



- Copy script dibawah ini pada **node.js**. Link sebagai berikut:
https://drive.google.com/drive/folders/1DRxEb-7F6pvjI7xqKHoZyS-RPr5N3_3c?usp=sharing

```
// =====
// WARTEG MANTAP JIWA BOT 🍽️
// Telegram Bot Version
// =====

require('dotenv').config();
const TelegramBot = require('node-telegram-bot-api');

// === MASUKKAN TOKEN TELEGRAM ANDA DI SINI ===
const token = process.env.TELEGRAM_BOT_TOKEN || 'diisi token dari telegram masing-masing mhs';

// Jalankan bot dengan polling
const bot = new TelegramBot(token, { polling: true });

// Error handling untuk bot
bot.on('error', (error) => {
  console.error('Bot error:', error);
});

bot.on('polling_error', (error) => {
  console.error('Polling error:', error);
});

// =====
// Data Menu Warteg
// =====
const menu = {
  "Ayam Goreng": 12000,
  "Ikan Lele Goreng": 10000,
  "Telur Dadar": 8000,
  "Sayur Asem": 5000,
  "Nasi Putih": 4000
};

// =====
// State Management
// =====
const userStates = new Map(); // chatId -> state
```

```
// =====
// WARTEG MANTAP JIWA BOT 🍽️
// Telegram Bot Version
// =====

require('dotenv').config();
const TelegramBot = require('node-telegram-bot-api');

// === MASUKKAN TOKEN TELEGRAM ANDA DI SINI ===
const token = process.env.TELEGRAM_BOT_TOKEN || 'diisi token dari telegram masing-masing mhs';

// Jalankan bot dengan polling
const bot = new TelegramBot(token, { polling: true });

// Error handling untuk bot
bot.on('error', (error) => {
  console.error('Bot error:', error);
});

bot.on('polling_error', (error) => {
  console.error('Polling error:', error);
});

// =====
// Data Menu Warteg
// =====
const menu = {
  "Ayam Goreng": 12000,
  "Ikan Lele Goreng": 10000,
  "Telur Dadar": 8000,
  "Sayur Asem": 5000,
  "Nasi Putih": 4000
};

// =====
// State Management
// =====
const userStates = new Map(); // chatId -> state
```

```

// =====
// Inline Keyboard Menu Utama
// =====
const mainMenu = {
  reply_markup: {
    inline_keyboard: [
      [{ text: "⌚ Lihat Menu", callback_data: "lihat_menu" }],
      [{ text: "🍽 Pesan Makanan", callback_data: "pesan_makanan" }],
      [{ text: "📍 Info Lokasi & Jam Buka", callback_data: "info_lokasi" }]
    ],
    [{ text: "📞 Hubungi Admin", callback_data: "hubungi_admin" }]
  }
};

// =====
// COMMAND /START
// =====
bot.onText(/\/start/, async (msg) => {
  const chatId = msg.chat.id;

  try {
    await bot.sendMessage(chatId, `⌚ Halo, selamat datang di *Warteg Mantap Jiwa*!
Saya *WartegBot*, siap bantu kamu pesan makanan enak dan hemat 😊

Silakan pilih menu di bawah ini:
`, { parse_mode: 'Markdown', ...mainMenu });
  } catch (error) {
    console.error('Error sending start message:', error);
  }
});

// =====
// HANDLER CALLBACK BUTTON
// =====
bot.on('callback_query', async (query) => {
  const chatId = query.message.chat.id;
  const action = query.data;

  try {
    switch (action) {
      // === FLOW 2: LIHAT MENU ===
      case 'lihat_menu':
        await bot.sendMessage(chatId, `⌚ *Menu Hari Ini:*
`);

        ⌚ Ayam Goreng - Rp12.000
        ⌚ Ikan Lele Goreng - Rp10.000
        ⌚ Telur Dadar - Rp8.000
        ⌚ Sayur Asem - Rp5.000
        ⌚ Nasi Putih - Rp4.000
    }
  }
});
```

```

        userStates.set(chatId, 'waiting_for_order');
        await bot.sendMessage(chatId, `

Baik! Ketik pesanan kamu ya, contoh:
*Pesan Ayam Goreng 1, Nasi 1, Sayur Asem 1*
    ` , { parse_mode: 'Markdown' });
    break;

// === FLOW 4: INFO LOKASI ===
case 'info_lokasi':
    await bot.sendMessage(chatId, `

⌚ Kami buka setiap hari pukul *07.00-21.00 WIB*
📍 Alamat: *Jl. Mawar No. 123, Cibubur*
❖ Google Maps: [bit.ly/warteg-mantapjiwa](https://bit.ly/warteg-
mantapjiwa)
    ` , { parse_mode: 'Markdown', ...mainMenu });
    break;

// === FLOW 5: HUBUNGI ADMIN ===
case 'hubungi_admin':
    await bot.sendMessage(chatId, `

Untuk pertanyaan atau pesanan besar (catering):
📞 *0812-3456-7890*
✉️ *wartegmantapjiwa@gmail.com*

Kami siap membantu Anda! 🌟
    ` , { parse_mode: 'Markdown', ...mainMenu });
    break;

default:
    await fallback(chatId);
}

await bot.answerCallbackQuery(query.id);
} catch (error) {
    console.error('Error handling callback query:', error);
    try {
        await bot.answerCallbackQuery(query.id, { text: 'Terjadi kesalahan,
silakan coba lagi.' });
    } catch (answerError) {
        console.error('Error answering callback query:', answerError);
    }
}
});

// =====
// HANDLER PESAN UMUM
// =====
bot.on('message', async (msg) => {
    const chatId = msg.chat.id;
    const userState = userStates.get(chatId);

    try {
        if (userState === 'waiting_for_order') {
            await handleOrder(msg);
        } else if (userState === 'waiting_for_confirmation') {
            await handleConfirmation(msg);
        } else if (userState === 'waiting_for_contact') {
            await handleContact(msg);
        }
    } catch (error) {

```

```

        console.error('Error handling message:', error);
        try {
            await bot.sendMessage(chatId, 'Terjadi kesalahan, silakan coba lagi
atau gunakan menu utama.', mainMenu);
        } catch (sendError) {
            console.error('Error sending error message:', sendError);
        }
    }
});

// =====
// FLOW 3.1: PROSES PESAN MAKANAN
// =====
async function handleOrder(msg) {
    const chatId = msg.chat.id;
    const orderText = msg.text;

    try {
        const items = parseOrder(orderText);
        const total = calculateTotal(items);

        if (total === 0) {
            userStates.delete(chatId);
            await bot.sendMessage(chatId, "Maaf, saya tidak menemukan item
valid dari pesanan kamu 😞", mainMenu);
            return;
        }

        const orderSummary = items.map(i => `${i.name}
x${i.quantity}`).join(" + ");
        userStates.set(chatId, 'waiting_for_confirmation');
        userStates.set(`#${chatId}_order`, { items, total, orderSummary });

        await bot.sendMessage(chatId, `
 Pesanan kamu: ${orderSummary}
⌚ Total: Rp${total.toLocaleString('id-ID')}`)

        Mau lanjut checkout? (ya/tidak)
        `, { parse_mode: 'Markdown' });
    } catch (error) {
        console.error('Error in handleOrder:', error);
        userStates.delete(chatId);
        await bot.sendMessage(chatId, "Terjadi kesalahan saat memproses
pesanank. Silakan coba lagi.", mainMenu);
    }
}

async function handleConfirmation(msg) {
    const chatId = msg.chat.id;
    const reply = msg.text.toLowerCase();

    try {
        if (reply === "ya") {
            userStates.set(chatId, 'waiting_for_contact');
            await bot.sendMessage(chatId, "Silakan kirim *nama dan nomor HP*
untuk konfirmasi pesanan.", { parse_mode: 'Markdown' });
        } else {
            userStates.delete(chatId);
            userStates.delete(`#${chatId}_order`);
        }
    } catch (error) {
        console.error('Error in handleConfirmation:', error);
        userStates.delete(chatId);
    }
}

```

```

        await bot.sendMessage(chatId, "Baik, pesanan dibatalkan. Silakan
lihat menu lain atau kembali ke menu utama.", mainMenu);
    }
} catch (error) {
    console.error('Error in handleConfirmation:', error);
    userStates.delete(chatId);
    userStates.delete(` ${chatId}_order`);
    await bot.sendMessage(chatId, "Terjadi kesalahan. Silakan coba
lagi.", mainMenu);
}
}

async function handleContact(msg) {
    const chatId = msg.chat.id;
    const contact = msg.text;

    try {
        userStates.delete(chatId);
        userStates.delete(` ${chatId}_order`);

        await bot.sendMessage(chatId, `

Terima kasih ${contact.split(" ")[0]}! 🎉
Admin kami akan segera menghubungi kamu untuk konfirmasi pesanan.

⌚ *Terima kasih sudah pesan di Warteg Mantap Jiwa!* 💕
`, { parse_mode: 'Markdown', ...mainMenu });
    } catch (error) {
        console.error('Error in handleContact:', error);
        await bot.sendMessage(chatId, "Terima kasih! Admin akan menghubungi
Anda.", mainMenu);
    }
}

// =====
// Fungsi Parsing & Hitung Total
// =====
function parseOrder(orderText) {
    const foundItems = [];
    const orderLower = orderText.toLowerCase();

    for (let key in menu) {
        const itemLower = key.toLowerCase();
        if (orderLower.includes(itemLower)) {
            // Cari jumlah item dengan regex
            const quantityRegex = new
RegExp(`(\d+)\s*${itemLower.replace(/\s+/g, '\\s+')}` , 'i');
            const match = orderText.match(quantityRegex);
            const quantity = match ? parseInt(match[1]) : 1;

            foundItems.push({
                name: key,
                price: menu[key],
                quantity: quantity
            });
        }
    }
    return foundItems;
}

function calculateTotal(items) {

```

```

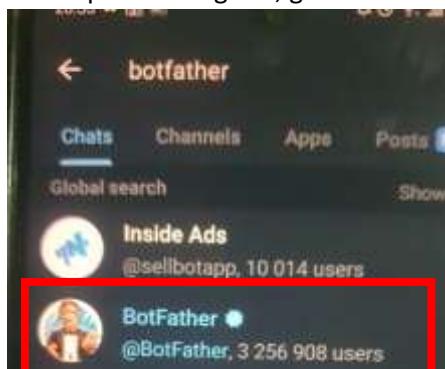
        return items.reduce((sum, item) => sum + (item.price * item.quantity),
    0);
}

// =====
// FLOW 6: FALBACK
// =====
async function fallback(chatId) {
    try {
        await bot.sendMessage(chatId, `Maaf, saya belum paham maksud kamu 😊
Silakan pilih menu berikut:
1 Lihat Menu
2 Pesan Makanan
3 Info Lokasi
4 Hubungi Admin
`, mainMenu);
    } catch (error) {
        console.error('Error in fallback:', error);
    }
}

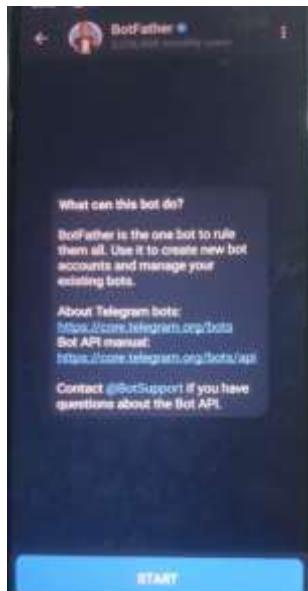
console.log("🤖 Warteg Mantap Jiwa Bot is running...");
```

6. Mendapatkan token telegram

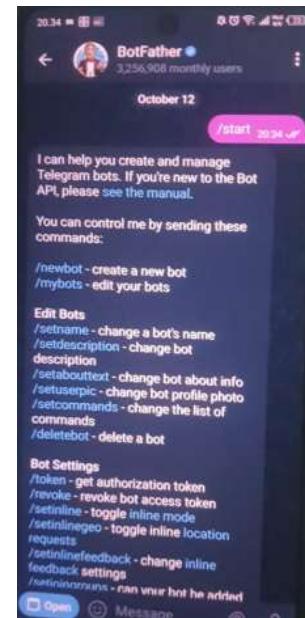
- Pastikan mahasiswa sudah install telegram di HP
- Sumber tutorial-> <https://www.shellhacks.com/how-to-get-telegram-bot-api-token-in-3-easy-steps/>
- Buka aplikasi telegram, gunakan fitur search : **Botfather**, dan klik



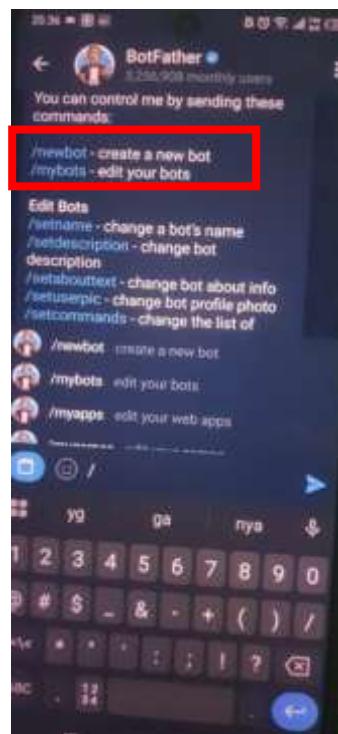
- Pilih Start



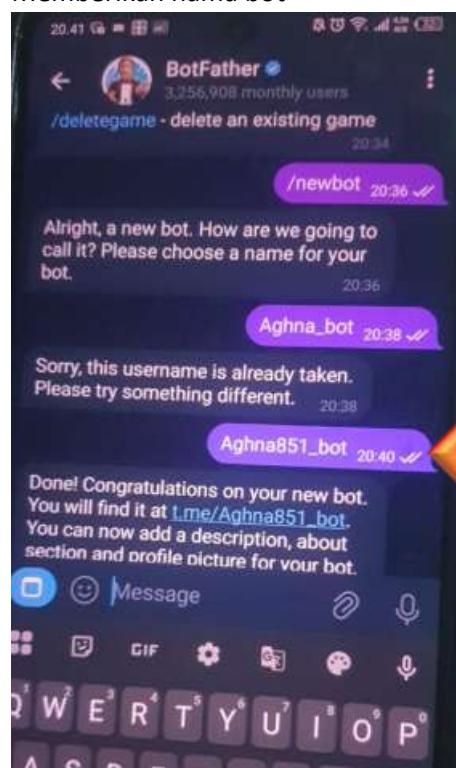
Output



- Create New Bot. Ketikan **/newbot**



- Memberikan nama bot

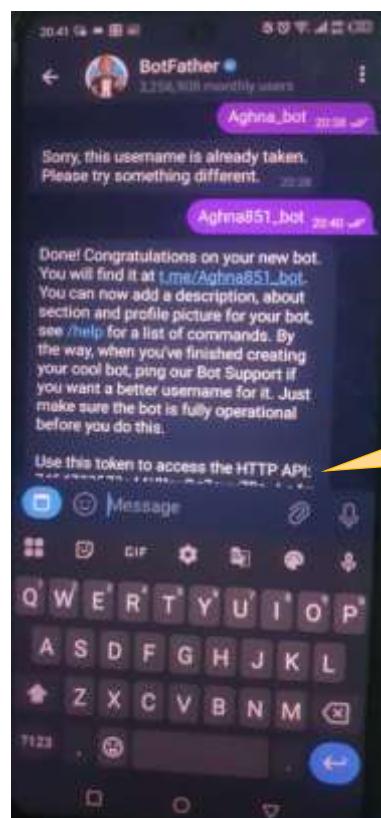


Jika pembuatan bot ditolak, ulangi prosesnya dan pastikan nama bot yang digunakan sudah sesuai dengan aturan penamaan dari Telegram

Aturan Penulisan Username (yang paling penting). Berikut syarat **WAJIB** dari Telegram:

Aturan	Penjelasan
Harus unik (belum pernah dipakai)	Tidak boleh sama dengan bot lain di seluruh Telegram. Jika sudah dipakai, BotFather akan menolak.
Harus diakhiri dengan kata "bot" atau "_bot"	Contoh yang benar: SriyadiBot, warteg_mantapjiwa_bot, sisteminformasi_bot
Tidak boleh ada spasi	Gunakan huruf dan angka saja (A-Z, 0-9, dan underscore _)
Minimal 5 karakter, maksimal 32 karakter	bot dihitung juga dalam panjang karakter
Case-insensitive	@SriyadiBot sama dengan @sriyadibot

- Pembuatan bot baru berhasil. Telegram secara otomatis memberikan token unik yang bersifat rahasia dan tidak boleh dibagikan kepada siapa pun. Berikut tampilan pesan pembuatan bot berhasil



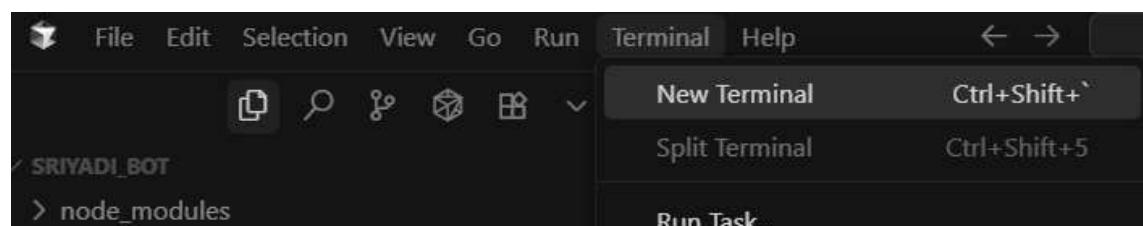
Token HTTP API: **745000000:AAHMAvRg7ovxZPz-Le4w7lhtdytrkkhdhydtjdujfuuhstydgtfa7**

Copy and paste token
di file bot.js

```
// === MASUKKAN TOKEN TELEGRAM ANDA DI SINI ===
const token = process.env.TELEGRAM_BOT_TOKEN ||
'745000000:AAHMAvRg7ovxZPz-Le4w7lhtdytrkkhdhydtjdujfuuhstydgtfa7';
```

pastikan save projek

7. Menjalankan bot telegram



Ketik: **bot.js**

```
Problems Output Debug Console Terminal Ports
PS D:\warteg-telegram-bot> node bot.js
[dotenv@17.2.3] injecting env (e) from .env -- tip: audit secrets and track compliance: https://datenow.com/ops
[Warteg Mantap Jiwa Bot is running...]
```

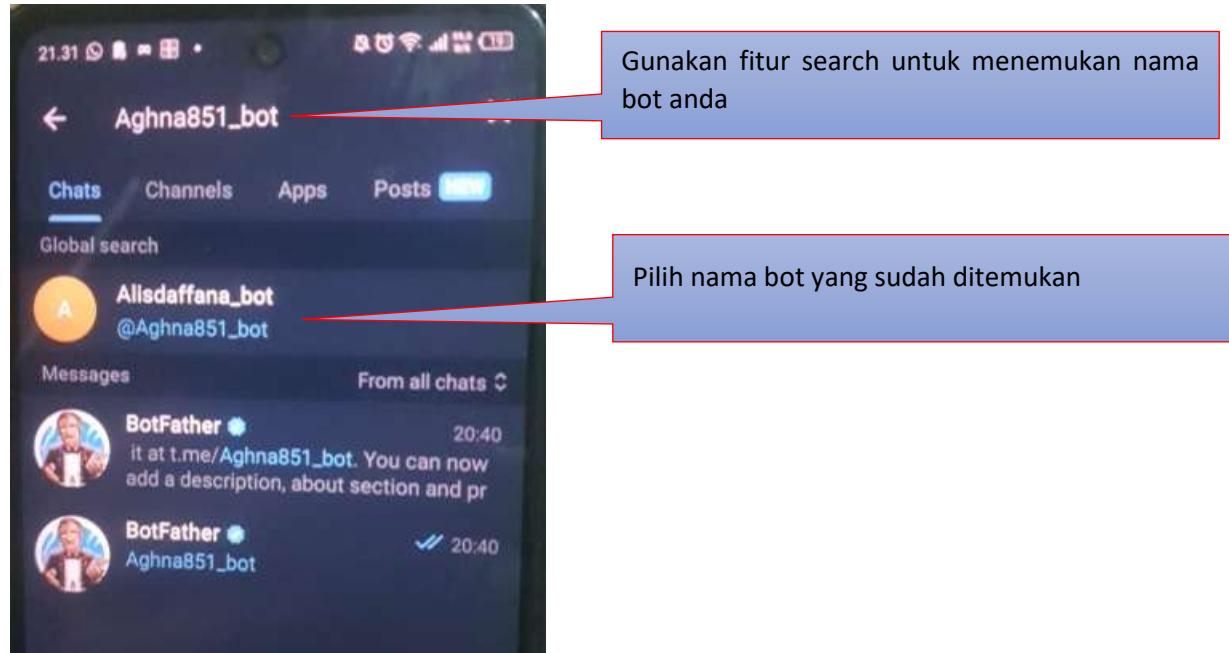
Aplikasi berjalan dengan baik

Untuk mematikan bot.js running, tekan keyboard CTRL+C

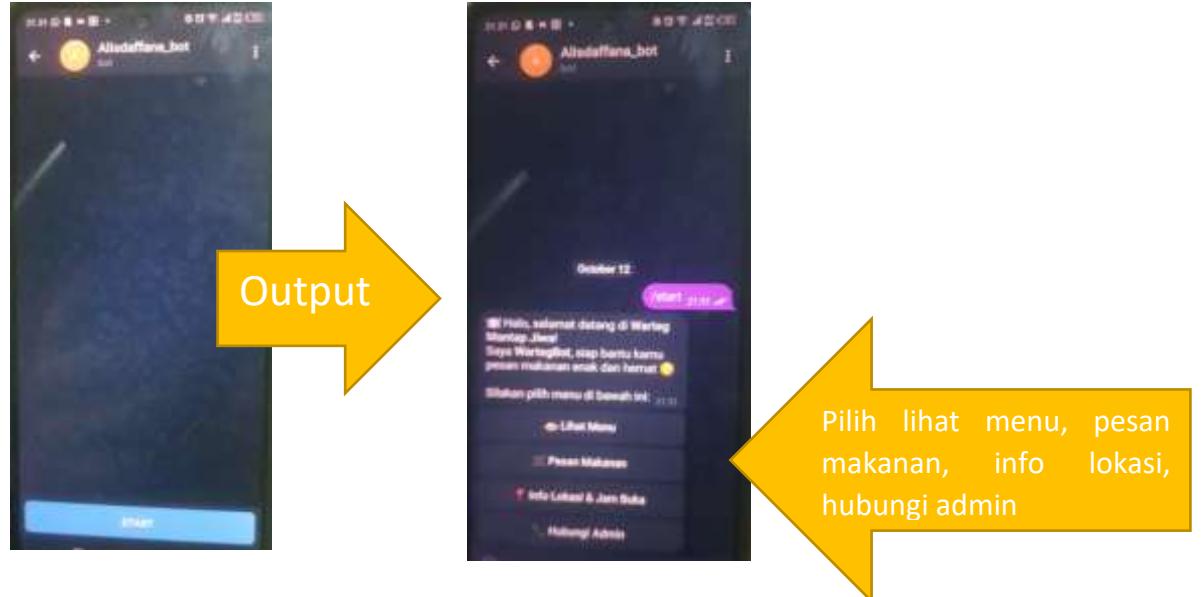
```
④ PS D:\Sriyadi_Bot> node bot.js
🤖 Warteg Mantap Jiwa Bot is running...
○ PS D:\Sriyadi_Bot>
```

8. Menguji bot telegram. Buka aplikasi telegram

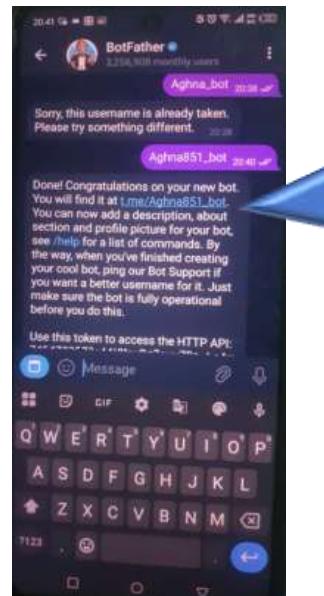
- Perhatikan pada gambar



- Klik Start sesuai gambar



9. Untuk membagikan bot tersebut ke banyak orang. Gunakan link yang diperoleh pada saat pembuatan nama bot berhasil. Perhatikan gambar



Bagikan ini ke orang lain:
https://t.me/Aghna851_bot

Tampilan Telegram Bot Aghna851_bot, di telegram orang lain



10. Mahasiswa dapat mengembangkan Git dan Hosting

Jika nanti ingin bot online 24 jam tanpa komputer harus nyala terus:

- Git: untuk versi kontrol & upload kode ke GitHub. <https://git-scm.com/>
- Hosting gratis: Render.com, Railway.app, Replit.com

11. Menu layanan masih terbuka untuk dikembangkan lebih lanjut. Mahasiswa dapat mengasah logika dan kreativitas melalui pembuatan bot Telegram, WhatsApp, maupun integrasi dengan ChatGPT dan teknologi AI lainnya