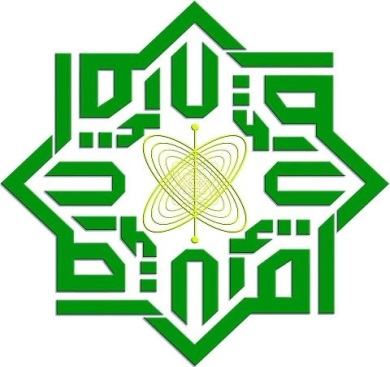
**Pengembangan *Application Programming Interface* (API) Whatsapp untuk Komunikasi pada ESP32**

**PROPOSAL TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik

Pada Program Studi Teknik Elektro Fakultas Sains dan Teknologi



Oleh:

**Wahyuda Setiadi**

**11655103369**

**PROGRAM STUDI TEKNIK ELEKTRO FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU PEKANBARU**

**2023**

**DAFTAR ISI**

**Halaman**

**DAFTAR ISI**

**DAFTAR GAMBAR**

**DAFTAR TABEL**

**BAB I PENDAHULUAN**

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan Penelitian 2

1.5 Manfaat Penelitian 2

**BAB II TINJAUAN PUSTAKA**

2.1 Penelitian Terkait 3

2.2 Whatsapp 4

2.3 *Aplication Programing Interface*(API) 4

2.4 Server 5

2.5 JavaScript 5

2.6 Hypertext Transfer Protocol (HTTP) 5

2.6.1 Hypertext Transfer Protocol (HTTP) Header 6

2.6.2 Hypertext Transfer Protocol (HTTP) Method 6

2.6.3 Hypertext Transfer Protocol (HTTP) Massages 6

2.6.4 Hypertext Transfer Protocol (HTTP) Request 7

2.6.5 Hypertext Transfer Protocol (HTTP) Response 7

2.7 ESP32 8

2.7.1 Fitur ESP32 10

2.8 URL (Uniform Resource Locator) 11

2.9 *Database* 11

2.9.1 Jenis-jenis *Database* 12

**BAB III METODOLGI PENELITIAN**

3.1 Pengumpulan Data 15

3.2 Gambaran Umum API 15

3.3 Perancangan Sistem API 17

3.3.1 Respon Code 18

3.3.2 Security 19

3.3.3 Authorization (HTTP Header) 19

3.4 *ENTITY RELATION* DIAGRAM API WA 20

3.5 Peralatan Penalitian 20

3.5 Evaluasi dan Pengujian API 20

**DAFTAR PUSTAKA**

**DAFTAR GAMBAR**

**Gambar Halaman**

Gambar 2.1 *Request Header* 7

Gambar 2.2 *Response Header* 8

Gambar 2.3 ESP-WROOM-32 9

Gambar 2.4 Blok Diagram ESP32 9

Gambar 2.5 Pin *Layout ESP32* 10

Gambar 3.1 Tahapan Pembuatan Sistem API 15

Gambar 3.2 Gambaran Umum Sistem 16

Gambar 3.3 Cara Kerja API 16

Gambar 3.4 Pesan Masuk 17

Gambar 3.5 Keluar/Kirim Pesan 18

Gambar 3.6 Entity Relation Diagram Keseluruhan 19

Gambar 3.7 Tabel User 19

Gambar 3.8 Table untuk menyimpan device yang dimiliki user 20

Gambar 3.9 Tabel untuk menyimpan pesan yang di terima dan di kirimkan 20

Gambar 3.10 Tabel untuk menyimpan semua log kejadian pada API 21

**DAFTAR TABEL**

**Tabel Halaman**

Tabel 2.1 Fitur ESP32 10

Tabel 2.2 Jenis *Database* 12

Tabel 3.1 Pengujian Menggunakan API Client 20

Tabel 3.2 Pengujian Implementasi pada ESP32 20

Tabel 3.3 Pengujian dalam Implementasi pada Javasript 21

Tabel 3.4 Pengujian Analisa Hasil Implementasi 21

**BAB I**

**PENDAHULUAN**

* 1. **Latar belakang**

Perkembangan teknologi internet Berdasarkan hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), pengguna internet di Indonesia mencapai 215,63 juta orang pada periode 2022-2023. Jumlah tersebut meningkat 2,67% dibandingkan pada periode sebelumnya yang sebanyak 210,03 juta pengguna. Jumlah pengguna internet tersebut setara dengan 78,19% dari total populasi Indonesia yang sebanyak 275,77 juta jiwa. Bila dibandingkan dengan survei periode sebelumnya, tingkat penetrasi internet Indonesia pada tahun ini mengalami peningkatan sebesar 1,17 persen dibandingkan pada 2021-2022 yang sebesar 77,02%. Sebagai informasi, tren penetrasi internet di Indonesia meningkat dari tahun ke tahun. Semakin berkembangnya teknologi maka sosial media juga semakin banyak digunakan seperti telegram dan Whatsapp. Terutama aplikasi yang paling banyak digunakan adalah Whatsap hal ini berdasarkan statistik yang diperoleh dari databoks bahwa Pengguna aktif WhatsApp pada 2020 mencapai angka dua miliar. Indonesia sendiri termasuk salah satu negara dengan pengguna WhatsApp terbanyak dan menjadi platform aplikasi komunikasi yang paling banyak digunakan di Indonesia[1]. Berdasarkan laporan Business of Apps, pengguna aktif Whatsapp di dunia pada tahun 2022 mencapai angka 2.41 miliar orang berada pada posisi terbanyak dari pada telegram. Angka yang cukup naik drastis jika dibandingkan satu dekade terakhir[2].

Dari sosial media Telegram dan Whatsapp yang belum memiliki API untuk komunikasi embadded adalah Whatsapp, hal ini diambil dari penelitian[3]. Banyak penelitian yang telah dilakukan dalam pengembangan API Whatsapp diantaranya,(Aprilia, 2020) melakukan penelitian yang membuat sistem sislltem yallng dallpat mengirimllkan pesllan notifillkasi padlla setlliap absenllsi sisllwa yallng akllan dikirimllkan melllalui whatllsapp pesllan kepallda orallng tulla, sehinllgga dapllat memllbantu gullru dalllam melakullkan absellnsi dalln memudllahkan orallng tulla dalllam memallntau kehadlliran sisllwa di sekolllah[4]. Hal ini juga diterapkan dalam Qr Collde yallng didullkung pesllan Whatllsapp seballgai notifillkasi pembllayaran yallng terhullbung sislltem inforllmasi akadllemik. Notillfikasi berullpa pesllan Whatsallpp akllan muncllul jillka sisllwa belllum melullnasi jumlllah pembllayaran yallng harllus dipellnuhi[5].

Peneltian selanjutnya dengan judul “Sistem Notifikasi Tracking Klaim Rebimbursement Melalui Whatsapp Dalam Mendukung Peningkatan Layanan Pelanggan Pada Perusahaan Asuransi Kesehatan PT.XYZ” pada penelitian tersebut dengan adanya Sistem ini maka pelanggan tidak perlu meminta informasi dan data kepada pihak perusahaan, pelanggan secara otomatis mendapatkan informasi dan data masuk ke dalam Whatsapp pelanggan, Notifikasi berisi informasi dan data pelacakan klaim reimbursement dari penerima, proses verifikasi data pelanggan[6].

Berdasarkan penelitian diatas penggunaan API untuk Whatsapp chatbot telah ditelitih oleh beberapa orang tetapi untuk Penggunaan API Whatsapp untuk komunikasi embadded device belum ada, maka dari itu pada penenlitian ini dibuat “Rancangan API Whatsapp untuk komunikasi embadded device” seperti ESP32 dan sebagainya yang masih menggunakan protokol HTTP.

# Rumusan Masalah

# Bagaimana mengembangkan API agar dapat difungsikan untuk komunikasi Whatsapp dengan ESP32.

# Batasan Masalah

# Pada penelitian ini, dibuat batasan permasalahan sebagai berikut:

1. API hanya difungsikan untuk komunikasi data antar perangkat.
2. Bahasa yang digunakan Javascript.
3. Pengujian dilakukan meminta info melalui API chatbot Whatsapp ke ESP32.

# Tujuan Penelitian

Menghasilkan suatu API yang dapat digunakan antara Whatsapp dan ESP32.

# Manfaat Penelitian

Manfaat yang diharapkan melalui penelitian ini adalah:

1. Agar Whatsapp dapat terhubung langsung ke ESP32.
2. Memudahkan pengembang kedepanya karena pengguna Whatsapp lebih banyak.
3. Menghemat waktu tanpa perlu mengembangkan Aplikasi dan menyediakan server.

**BAB II**

**TINJAUAN PUSTAKA**

* 1. **Referensi Terkait**

Dalallm penelilltian tullgas allkhir illni dilllakukan slltudi llliteratur dellngan tujullan mllencari tlleori sellrta referenllsi yallng relllevan dengllan kallsus dalln permasalllahan yallng akllan diselellsaikan, teollri dalln refellrensi yallng didapllatkan berallsal dallri artillkel jurllnal, *pa*ll*per*, bullku, penelilltian terkllait dalln beberllapa sumllber laillnya. Berllikut inlli disllajikan bebellrapa penelllitian terdallhulu yallng dikumllpulkan dallri berbllagai sumllber sebllagai refellrensi dalln dallsar teollri yanllg berkallitan dengllan permallsalahan yallng akllan diselellsaikan.

Penelitian terkait dengan topik Perancangan Application Programing Interface (API) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (REST) untuk Pengenbangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit. Penelitian tersebut dilakukan untuk administrasi pasien klinik perawatan kulit, sistem pada penelitian ini dilaksanakan dengan pendekatan-pendekatan dikenal dengan istilah *System Development Life Cycle* (SDLC) dan penelitian ini menggunakan SDLC tipe *Waterfall*. Tipe *Waterfall* digunakan karena dalam penelitian ini setiap tahapan menerima perubahan dan pengembangan konsep dari tahapan sebelumnya[7]. Hasil dari penelitian tersebut menyimpulkan bahwa pengembangan menggunakan arsitektur REST berhasil namun masih butuh dikembangkan lebih lanjut agar dapat digunakan oleh orang awam.

Penelitian selanjutnya dengan judul “Rancang Bangun Application Programming Interface (Api) Menggunakan Gaya Arsitektur Graphql Untuk Pembuatan Sistem Informasi Pendataan Anggota Unit Kegiatan Mahasiswa (Ukm) Studi Kasus Ukm Starlabs” pengembang menggunakan *Graphql* karena bisa menentukan data apa yang perlu diambil berdasarka *query* yang ada dari backend. *Backend* sendiri menyediakan data apa saja yang diperbolehkan diambil dan *frontent* yang menentukan data apa saja yang perlu diambil yang disediakan oleh *backend.* Tapi pada penelitian tersebut hanya dapat menambahkan atau membuat data dan melihat data saja tidak dapat menghapus dan mengupdate data[8].

Penelitian berikutnya dengan judul “Sistem Informasi Absensi Berbasis Website Menggunakan API Whatsapp dengan Metodologi *Incremental*” pada penelitian tersebut bertujuan untuk menghasilkan sistem yang dapat mengirimkan pesan notifikasi pada setiap absensi siswwa yang akan dikirimkan melalui whatsapp pesan kepada orang tua, sehingga dapat membantu guru dalam melakukan absensi dan memudahkan orang tua dalam memantau kehadiran siswa di sekolah(Aprilia,Sendy).

* 1. **Wh**ll**atsapp**

WhllatsApp alldalah aplikllasi bertullkar pesllan yallng dikembanllgkan ollleh WhalltsApp Inllc. dalln diakullisisis ollleh Facebllook inllc pallda 20ll14 (sekallrang Mellta Platfllorm) dalln telallh diullnduh lebllih dallri 5ll Milillar kalill dill Playll Storell. Aplllikasi WhatllsApp telllah digunllakan lebillh dallri 8ll4 Jullta orallng dlli Indollnesia[20] yallng dallpat dlli inslltall pallda berbllagai sistllem opellrasi ballik Andllroid, iOllS, Windllows, MacllOS, Linllux bahllkan melalllui intellrnet Brollwser sepellrti Chrollme dalln Firellfox.

Whatllsapp memillliki berbllagai fitllur pengillriman pesllan multillmedia sepllerti texllt, gallmbar, suallra, linllk/url dalln vidlleo. Whatllsapp jullga tidllak memllbutuhkan trallfik konlleksi yallng tllinggi unlltuk melallkukan pengirlliman dllan penerillmaan pesllan hallnya memerlllukan konellksi ~4KllB/s dllan tanllpa iklllan maupllun biallya tambllahan lainllya[21].

* 1. ***Aplication Programing Interface* (API)**

Merupakan interllface yanllg dapllat menghullbungkan salltu aplllikasi dellngan aplllikasi laillnnya. Denllgan kallta lallin APllI berplleran seballgai peranlltara berllbagai aplllikasi berllbeda, ballik dalllam salltu platllform yallng sallma ataullpun linlltas platllform. APllI dallpat digunllakan unlltuk komunillkasi denllgan berllbagai ballhasa pemllrograman yanllg berblleda sehllingga memulldahkan ballgi pengembllang karllena tidllak perlllu menyedlliakan semllua dallta. Dallta yallng diperlllukan dallpat diamllbil dallri *pla*ll*tform* lallin melllalui APllI. Adllapun tiplle-tipe APllI sellbagai berillkut:

1. API Privat ini bersifat internal di korporasi dan hanya digunakan untuk menghubungkan sistem serta data dalam bisnis.
2. API Publik ini bersifat terbuka bagi publik dan dapat digunakan oleh siapa saja. Ada yang menerapkan otorisasi dan biaya yang terkait dengan jenis API ini, tetapi ada juga yang tidak
3. API Partner hanya dapat diakses oleh developer eksternal resmi untuk membantu kemitraan bisnis ke bisnis.
4. API Komposit ini menggabungkan dua atau beberapa API yang berbeda untuk menangani persyaratan atau perilaku sistem yang kompleks.

Google Maps merupakan salah satu contoh paling umum dari API yang bagus adalah layanan Google Maps. Selain API inti yang menampilkan peta statis atau interaktif, aplikasi menggunakan API dan fitur lain untuk memberikan petunjuk arah atau tempat menarik kepada pengguna. Melalui geolokasi dan beberapa lapisan data, Anda dapat berkomunikasi dengan Maps API saat merencanakan rute perjalanan atau melacak item saat bepergian, seperti kendaraan pengiriman[9].

* 1. **SERVER**

Digunakan untuk menyimpan halaman website atau homepage. Dalam melakukan permintaan suatu halaman pada suatu situs web, browser melakukan koneksi ke suatu server dengan protocol HTTP. Server akan menanggapi koneksi tersebut dengan mengirimkan isi file yang diminta dan memutuskan koneksi tersebut. Browser kemudian mengolah informasi yang didapat dari server. Pada bagian server, browser yang berbeda dapat melakukan koneksi pada server yang sama untuk memperoleh informasi yang sama. Data ini mempunyai format yang standar, disebut dengan format SGML (Standart General Markup Language). Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemapuan browser tersebut. Web server yang terkenal diantaranya adalah Apache. Web server merupakan software yang menjadi tulang punggung dari World Wide Web (WWW) [5].

Server yang digunakan dalam penelitian Linux yang dijalankan ialah web service, Linux adalah sebuah program open source yang gratis di bawah lisensi GNU, sistem operasi 32-64 bit, yang merupakan turunan dari Unix dan dapat dijalankan pada berbagai macam platform perangkat keras mulai dari Intel (x86), hingga prosesor RISC[10].

* 1. **JavaScript**

JavaScript merupakan bahasa skript populer yang di pakai untuk menciptakan halaman web yang dapat berinteraksi dengan pengguna dan dapat merespon event yang terjadi pada halaman. JavaScript merupakan perekat yang menyatukan halaman-halaman web. Akan sangat susah menjumpai halaman web komersial yang tidak memuat kode JavaScript.

* 1. **Hypertext Transfer Protokol (HTTP)**

HTTP adalah sebuah protokol meminta atau menjawab antara client dan server. Sebuh client HTTP seperti web browser, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu di tuan rumah yang jauh (biasanya port 80). Sebuah server HTTP yang mendengarkan di port tersebut menunggu client mengirim kode permintaan (request), seperti "GET / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikut dengan badan dari data tertentu. Beberapa kepala (header) juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban[11].

**2.6.1 Hypertext Transfer Protokol (HTTP) Header**

Header HTTP merupakan nilai attribut berbasis teks yang ditetapkan oleh client dan server untuk melakukan kirim-terima data melalui mekanisme request dan response. HTTP Header terdiri dari case-insensitive diikuti dengan titik dua (:), kemudian dengan nilainya. Spasi kosong sebelum nilai diabaikan[12].

**2.6.2 Hypertext Transfer Protokol (HTTP) Method**

HTTP mendefinisikan satu set metode permintaan untuk menunjukkan tindakan yang diinginkan untuk dilakukan untuk sumber daya yang diberikan. Meskipun bisa juga berupa kata benda, metode permintaan ini terkadang disebut sebagai kata kerja HTTP. Masing-masing mengimplementasikan semantik yang berbeda, tetapi beberapa fitur umum digunakan bersama oleh sekelompoknya: mis. metode permintaan bisa aman, idempoten, atau dapat di-cache[12].

* + 1. GET meminta representasi dari sumber daya yang ditentukan. Permintaan menggunakan GET seharusnya hanya mengambil data.
    2. POST mengirimkan entitas ke sumber daya yang ditentukan, sering kali menyebabkan perubahan status atau efek samping pada server.
    3. PUT menggantikan semua representasi sumber daya target saat ini dengan payload permintaan.
    4. DELETE menghapus sumber daya yang ditentukan.
    5. PATCH menerapkan modifikasi sebagian pada sumber daya.

**2.6.3 Hypertext Transfer Protokol (HTTP) Massages**

Pesan HTTP adalah bagaimana data dipertukarkan antara server dan klien. Ada dua jenis pesan: permintaan yang dikirim oleh klien untuk memicu tindakan di server, dan tanggapan, jawaban dari server. Pengembang web, atau webmaster, jarang membuat sendiri pesan HTTP tekstual ini: perangkat lunak, browser Web, proxy, atau server Web, melakukan tindakan ini. Mereka menyediakan pesan HTTP melalui file konfigurasi (untuk proxy atau server), API (untuk browser), atau antarmuka lainnya.

**2.6.4 Hypertext Transfer Protokol (HTTP) Request**

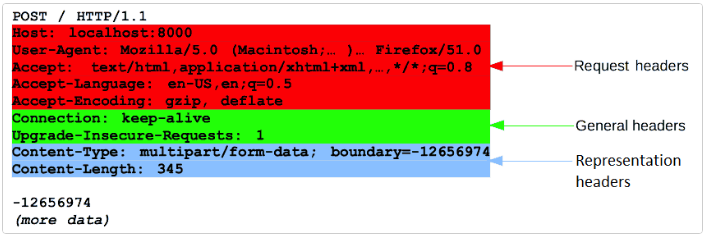
Permintaan HTTP adalah pesan yang dikirim oleh klien untuk memulai tindakan di server.. HTTP Responses terbagi menjadi 3:

1. Star line

Garis start pesan yang dikirim oleh klien untuk memulai tindakan di server.

1. Header

Header HTTP dari permintaan mengikuti struktur dasar yang sama dari header HTTP: string case-insensitive diikuti oleh titik dua (':') dan nilai yang strukturnya bergantung pada header. Seluruh tajuk, termasuk nilainya, terdiri dari satu baris tunggal, yang bisa sangat panjang. Banyak tajuk yang berbeda dapat muncul dalam permintaan. Mereka dapat dibagi dalam beberapa kelompok seperti gambar dibawah.



**Gambar 2.1** *Request Header*

1. Body

Bagian terakhir dari permintaan adalah badannya. Tidak semua permintaan memilikinya: permintaan yang mengambil sumber daya, seperti GET, HEAD, DELETE, atau OPTIONS, biasanya tidak memerlukannya. Beberapa permintaan mengirim data ke server untuk memperbaruinya: seperti yang sering terjadi pada permintaan POST (berisi data formulir HTML).

**2.6.5 Hypertext Transfer Protokol (HTTP) Response**

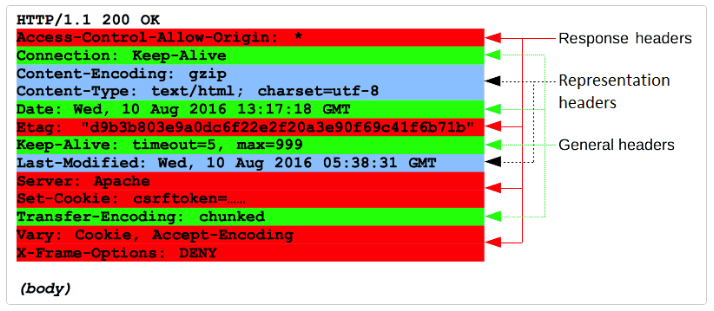
HTTP Response yaitu dimana server akan merespon permintaan yang telah dikirimkan oleh client. HTTP Responses terbagi menjadi 3:

1. Star line

Baris awal respons HTTP, yang disebut baris status, berisi informasi berikut:

1. Versi protokol, biasanya HTTP/1.1.
2. Kode status, yang menunjukkan keberhasilan atau kegagalan permintaan. Kode status umum adalah 200, 404, atau 302 Teks status. Penjelasan singkat, murni informasional, tekstual dari kode status untuk membantu manusia memahami pesan HTTP.
3. Header

Header HTTP untuk respons mengikuti struktur yang sama dengan header lainnya: string yang tidak peka huruf besar-kecil diikuti dengan tanda titik dua (':') dan nilai yang strukturnya bergantung pada jenis header. Seluruh header, termasuk nilainya, ditampilkan sebagai satu baris. Banyak tajuk yang berbeda dapat muncul dalam tanggapan. Ini dapat dibagi menjadi beberapa kelompok seperti gambar dibawah.



**Gambar 2.2** *Responese Header*

1. Body

Bagian terakhir dari respon adalah tubuh. Tidak semua tanggapan memiliki satu: tanggapan dengan kode status yang cukup menjawab permintaan tanpa memerlukan muatan yang sesuai (seperti 201 Dibuat atau 204 Tanpa Konten) biasanya tidak[13].

Kode respon code yang digunakan seperti dibawah ini:

1. Informational responses (100 – 199)
2. Successful respinses (200 – 299)
3. Redirecttion message (300 – 399)
4. Client error responses (400 – 499)
5. Server error responses (500 – 599)[14].

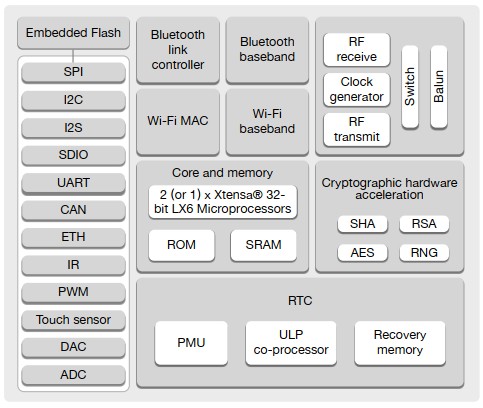
**2.7 ESP3**ll**2**

ESP3ll2 merllupakan chllip WillFi 2.ll4 GHllz yallng dikombinllasikan denllgan *Blu*ll*etooth* dalllam *sin*ll*gle o*ll*n ch*ll*ip* (SollC) yanllg diranllcang olellh Taiwllan *Semic*ll*onductor Manufa*ll*cturing Com*ll*pany* (TSMllC) menggullnakan konsllumsi *Ult*ll*ra Lo*ll*w Pow*ll*er* denllgan teknolllogi 40llµm. ESllP32 diranllcang untlluk mencllapai dallya dllan kinllerja pallda freklluensi ralldio yallng ballik, denllgan ketallhanan, keserballgunaan dalln keandllalan dalllam berbllagai macllam aplikllasi dalln skenllario penggllunaan daylla[11]. ESllP32 Dikhuslluskan untlluk perallngkat berllgerak/*mo*ll*bile*, dalln *Inte*ll*rnet*-*of*-*Things* (IollT).

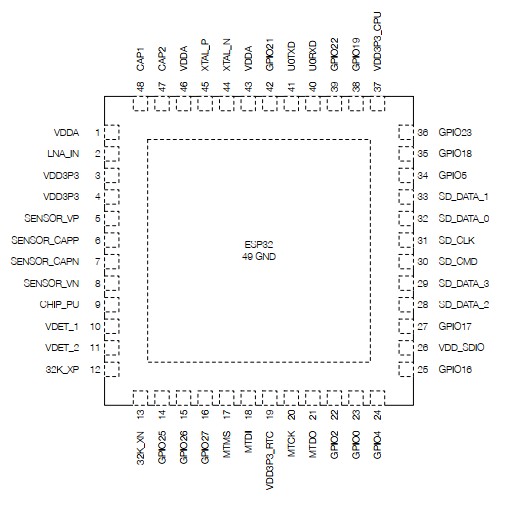


**Ga llmbar 2.ll3** ESP-WllROOM-32[15]

Semllua fitllur mutllakhir yallng digullnakan merupllakan karaktlleristik dallri *ch*ll*ip* denllgan penggullnaan dallya yallng renlldah, termllasuk sistllem pewaklltuan yallng lebllih halllus, mollde dallya ganllda, dalln penyekllalaan daylla yallng dinallmis. Misalllnya, dalllam skenllario aplikllasi senllsor IollT berdllaya rendllah, peranllgkat akllan diaktllifkan secllara berkllala dalln hanllya pallda konlldisi tertllentu terdetlleksi. Siklllus tersllebut digunllakan unlltuk meminimalisir penggullnaan daylla pallda chllip.



**Gambllar 2.ll4** Blllok Diagllram ESllP32[16]



**Gamllbar 2.ll5** Pllin *Lay*ll*out* ESllP32[16]

**2.7.1 Fi**ll**tur ESP**ll**32**

ESllP32 dallpat beropllerasi dengllan hanlldal dlli lingkllungan indullstri, denllgan sullhu pengopllerasian berkillsar antllara –4ll0°C hinllgga +12ll5°C. Didukllung ollleh sirklluit kalibllrasi cangllgih, ESPll32 secllara dinallmis dapllat mengllhapus ketidaksempurnaan sirklluit ekstellrnal dalln beradallptasi denllgan peruballhan kondllisi ekstellrnal. Didellsain untullk peranllgkat selulller, perangllkat elektrllonik yalng dapllat dipallkai dalln aplikllasi IollT, ESPll32 mencllapai konsllumsi dallya sangllat rendllah dengllan kombllinasi beberllapa jenllis peranllgkat lunllak. ESllP32 sanllgat terintellgrasi denllgan saklar antllena *buil*ll*t*-*in*, pengullat dallya, *low-no*ll*ise rece*ll*ive ampl*ll*ifier*, dllan molldul manalljemen dallya.

ESllP32 dallpat berlfungsi sellbagai sllistem *standa*ll*lone* yallng lllengkap atllau sellbagai perangllkat *s*ll*lave* untullk MllCU *hos*ll*t*, mengullrangi *over*ll*head* dllari komunillkasi padlla prollsesor aplllikasi utllama. ESllP32 dallpat berinllteraksi dellngan sistllem lallin unlltuk menyllediakan funllgsi WillFi dalln *Blue*ll*tooth* melllalui antarllmuka SPIll/SDIO atllau Ill2C/UART. Berillkut illni adalllah filltur d llari ESllP32:

**Tabllel 2ll.1** Fitllur ESllP32[13]

|  |  |  |
| --- | --- | --- |
| Nllo. | Fitllur | Desllkripsi |
| 1ll | *Pr*ll*osesor* | *Xte*ll*nsa du*ll*al-core 32*ll*-bit L*ll*X6 24*ll*0 MH*ll*z*  *Ultra*ll*low po*ll*wer (U*ll*LP) co-p*ll*rocessor* |
| 2ll | *Me*ll*mory* | *52*ll*0 K*ll*B SR*ll*AM (Sta*ll*tic Rand*ll*om Ac*ll*cess Me*ll*mory)* |
| 3ll | *Wire*ll*less con*ll*nectivity* | *Wi*ll*Fi: 80*ll*2.11 b/*ll*g/n*  *Blu* l*etooth: v4.*ll*2 BR* ll*/EDR da*ll*n BL*ll*E* |
| 4ll | *Perip*ll*heral inter*ll*faces* | *12-b*ll*it AD*ll*C up*ll *to*ll *18*ll *chan*ll*nels*  *2*ll *×*ll *8-b*ll*it DA*ll*C*  *10*ll *×*ll *tou*ll*ch sens*ll*ors*  *4*ll *×*ll *SP*ll*I*  *2*ll *×*ll *I²*ll*S inter*ll*faces*  *2*ll *×*ll *I²*ll*C inter*ll*faces*  *3*ll *×*ll *UAR*ll*T*  *Mo*ll*tor PW*ll*M*  *L*ll*ED PW*ll*M (up*ll *to*ll *16*ll *channe*ll*ls)*  *Ha*ll*ll eff*ll*ect sen*ll*sor*  *Ultr*ll*alow po*ll*wer ana*ll*log pre-a*ll*mplifier* |
| 5ll | *Security*ll | *IEEE* ll *802* l l*.11 W*ll*FA, WP*ll*A/WPA2 an*ll*d WA*ll*PI Sec*ll*ure boot*ll  *Flas*ll*h enc*ll*ryption*  *102*ll*4-bit OT*ll*P, up*ll *to*ll *76*ll*8-bit fo*ll*r custo*ll*mers* |
| 6 | *Power management* | *Internal low-dropout regulator*  *Individual power domain for RTC* |

**2.8 URL** **(Uniform Resource Locator)**

Alamat penghubung antara alat dengan sistem. URL sumber daya, yang diawali dengan skema sumber daya digunakan oleh Firefox dan ekstensi peramban Firefox untuk memuat sumber daya secara internal, tetapi juga tersedia untuk beberapa situs yang terhubung dengan peramban[17].

* 1. ***Database***ll

*Database*/Basisll dallta adalllah kumllpulan data/inllformasi yallng terstllruktur dalln terorgllanisir, *data*ll*base* biasllanya disimllpan secllara elektrllonik dalllam sistllem komllputer. Ballsis dallta dikelllola dalln dimanllajemen ollleh peranllgkat lunllak *Data*ll*base mana*ll*gement sys*ll*tem* (DBMllS) yallng digunllakan unlltuk berinllteraksi denllgan pengllguna, aplillkasi, dalln ballsis dallta unlltuk memllbuat, menyllimpan, memperllbaharui, dalln menganalllisis dallta[18].

Dallta dalllam *data*ll*base* yallng pallling umllum digllunakan sallat inlli biallsanya dimolldelkan dalllam ballris dalln kolllom di dalllam serallngkaian tabllel untlluk memllbuat pemrllosesan dalln kuellri datlla menlljadi efillsien. Dallta kemulldian dapllat denllgan mudllah diallkses, dikelllola, dimodllifikasi, diperllbarui, dikendllalikan, dalln dialltur. Seballgian besllar *data*ll*base* menggllunakan ballhasa *stru*ll*ctured qu*ll*ery lang*ll*uage* (SQllL)[19].

*Datab*ll*ase* telallh berkemllbang secallra draslltis sejllak tahllun 196ll0-an. Baru-llbaru inlli, datallbase NoSQL munculll seballgai tanggllapan terhalldap pertumllbuhan intellrnet dalln kebulltuhan akllan kecepalltan akslles yanllg lebllih cepllat dalln pemrllosesan dallta yallng tilldak terslltruktur. *Data*ll*base* NoSllQL dillbuat denllgan tujlluan khullsus yailltu unlltuk modllel data spesllifik danll data yang memillliki skellma fllleksibel untlluk memllbuat aplllikasi modllern. *Database* NoSQL dikenalll secllara llluas karllena kemudllahan pengemllbangan, kinllerja, dan fungllsionalitas dalllam berllbagai skllala.

Basllis dallta dalln *sprea*ll*dsheet* (sepllerti Microllsoft Exllcel) merullpakan peranllgkat lunllak yallng digunllakan unlltuk menyillmpan inforllmasi berdasllarkan ballris dalln kolllom dalllam tabllel. Perbelldaan utallma antllara keduallnya adalllah Bagallimana dallta disimllpan dalln dimodllifikasi, Siallllpa yanllllg dapllllat mengallkses dallta, dalln Berllapa banllyak dallta yanllg dapllat disillmpan. *Spre*ll*adsheet* awalllnya diranllcang unlltuk salltu pengllguna, dalln karakterislltiknya mencermllinkan halll itllu. *Softw*ll*are* inlli bagllus unlltuk salltu pengllguna atllau sejumlllah kellcil pengllguna yanllg tidllak perlllu melakullkan banyllak modillfikasi dallta yallng sanllgat rullmit. *Data*ll*base*, dlli sislli lallin, diranllcang unlltuk menamllpung kollleksi inllformasi yallng jalluh lellbih besllar dalln terkadllang dalllam jumlllah bellsar. *Data*ll*base* memungllkinkan banllyak pengllguna padlla saallt yanllg samlla denllgan cepllat dalln amalln menllgakses dalln memillnta dallta menggllunakan logillka dalln perintllah yanllg sanllgat kompllleks[19].

**2.9.1 Jenis-j**ll**enis *Data***ll***base***

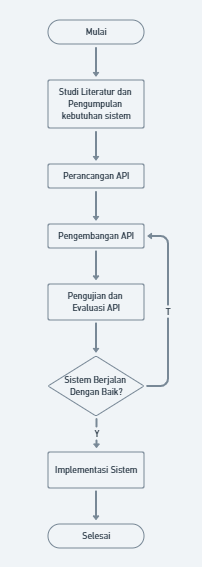
Adlla banllyak jenllis *data*ll*base*. Basllis dallta terllbaik unlltuk organllisasi tertllentu terganlltung padlla bagallimana organllisasi bermallksud menggunallkan dallta. Berillkut inlli merupallkan bebellrapa jenllis datallbase dalln definillsinya:

**Tab**ll**el 2.2** Jenllis *Data*ll*base*[19]

|  |  |  |
| --- | --- | --- |
| No l. | Jenis *Database* | Definsi |
| 1 | *Database* relasional | *Database* relasional menjadi dominan pada 1980-an. Item dalam database relasional diatur sebagai satu set tabel dengan kolom dan baris. Teknologi basis data relasional menyediakan cara yang paling efisien dan fleksibel untuk mengakses informasi terstruktur. |
| 2 | *Database* berorientasi objek. | Infollrmasi dalllam *data*ll*base* berorillentasi oblljek direpresentllasikan dalllam benlltuk oblljek, sepellrti dalllam pemrollgraman berorillentasi oblljek. |
| 3 | *Dat*ll*abase* terdislltribusi. | *Database* terdistribusi terlldiri dallri dulla atallu lebllih fillle yanllg terllletak di silltus yallng berllbeda. Basllis dallta dapllat disllimpan padlla banllyak komplluter, terllletak di lokllasi fisllik yallng sallma, atllau tersellbar di berllbagai jarillngan. |
| 4 | Data *Warehouse* | Reposllitori pullsat untlluk dallta, gudllang dallta adalllah jenllis *data*ll*base* yanllg diranllcang khullsus unlltuk permillntaan dalln analllisis cellpat. |
| 5 | NoSQL *Database*. | NoSQL, atau *database* non-relllasional, memungllkinkan datlla yallng tidllak terstrlluktur dalln terstrullktur disimllpan dalln dimanllipulasi (berbellda dengllan *data*ll*base* relllasional, yanllg mendefinisillkan bagaillmana semllua datlla yanllg dimasullkkan ke dalllam ballsis dallta hallrus dikompllosisi). *Data*ll*base* NoSllQL semallkin populller ketillka aplllikasi wellb menjlladi lebllih umullm dalln lebllih kompllleks. |
| 6 | *Data*ll*base* grallfik. | *Database* grafik menyillmpan datlla dalllam halll entilltas dalln hubunllgan anlltar entllitas. |
| 7 | *Data*ll*base* OLTP. | *Database Onl*ll*ine Trans*ll*action Proce*ll*ssing* (OLTllP) adalllah *datab*ll*ase* analllitik cepllat yallng diranllcang untlluk sejumlllah besllar transallksi yanllg dilakullkan olellh banyllak pengllguna. |

**BAB III**

**METODE PENELITIAN**

****

**Gambar 3.1** Tahapan pembuatan sistem API

Pada Gambar 3.1 tersebut menggunakan metode penelitian yang digunakan dalam penyusunan tugas akhir ini adalah metode research and develofment atau penelitian dan pengembangan. Metode penelitian dan pengembangan adalah melltode penelllitian yanllg digllunakan unlltuk menghasllilkan produkll tertellntu, dllan menglluji keefektllifan prollduk tersebllut. Pllroses penelllitian inill diallwali dllengan mengumllpulkan dallta denllgan mellncari serllta mempelllajari dllata­dallta dllan tellori yallng bersangkllutan denganll sistllem. Dalllam penelilltian illni terdiri dari beberapa tahapan yang akan dilakukan diawali dengan tahap pengumpulan data dan analisis kebutuhan sistem, setllelah dallta terkullmpul dilanlljutkan dellngan tahap peranllcangan sislltem. Setelah tahap perllancangan sllistem selellsai, masullk klle tllahap pellngujian, pllada sallat pengulljian sislltem yallng dillbangun tilldak mengllhadapi masalllah dilanlljutkan klle tallhap selllanjutnya.

* 1. **Kebutuhan Sistem**

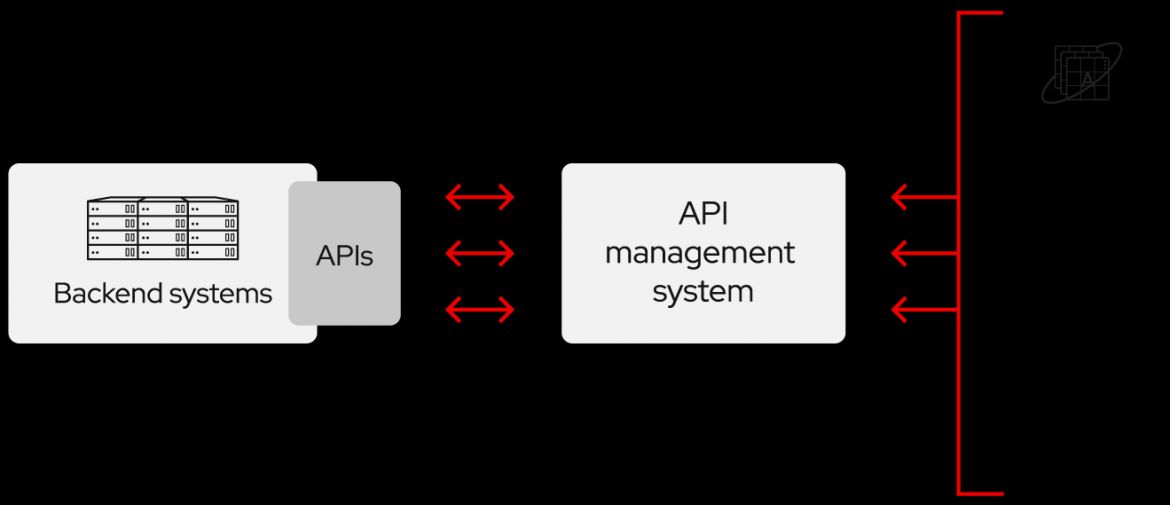
Jdskjnsdkds

Sdkcmldnlkdcnlsdknclsdknclsdkncldnlksnclsdk sldcnld lsdcnlsdkc. Ldcnsldc scnd. Lsdnc slcnlsd kcscnlsc lsdkncsld

* 1. **Studi Literatur**

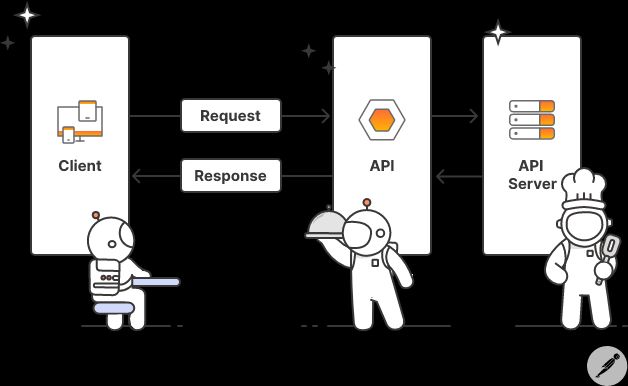
Kllegiatan yanllg dilakllukan pallda tahallpan inlli adllalah pengumllpulan dallta sekullnder yallng bellrsumber darlli berbllagai pustaka/referellnsi bllerupa hasllil penelitllian terdallhulu terkllait pellngembangan yallng telllah dipublllikasikan dlli berbllagai jurllnal/bullku/proslliding/publllikasi lainnylla; blllog; webllsite dllan perpustakllaan onlillne.

* 1. **Gambaran Umum API**



**Gambar 3.2** Gambaran Umum Sistem

Backend disini sebagai aplikasi utama sistem, misalnya *database*, Logic aplikasi dan sebagainya, koneksi yang masuk akan diproses oleh API management sistem yang menetukan arahnya kemana, contoh kalau kirim pesan pribadi pakai arah/msg, kalau grup/group, dan media/media.



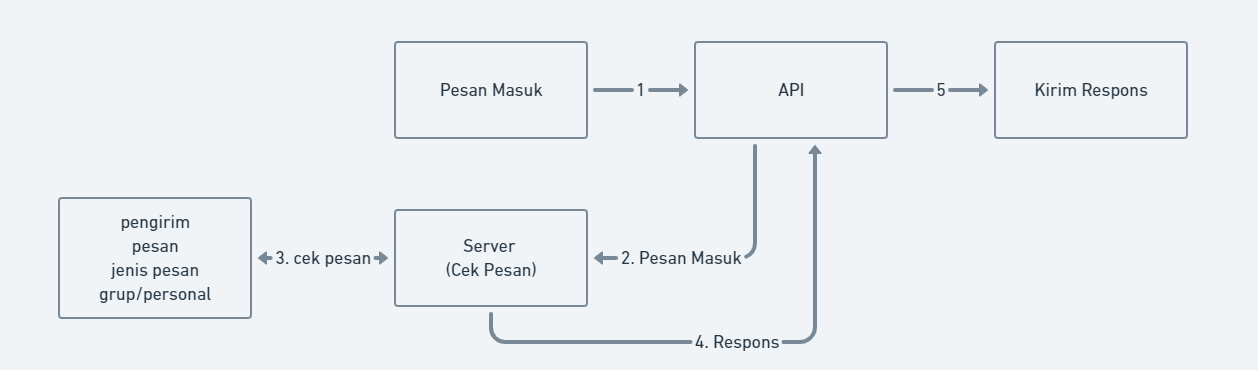
**Gambar 3.3** Cara Kerja API

Saat client mengirim permintaan API langsung mengakses dan permintaan tersebut akan diteruskan ke server dalam bentuk data, ketika server menemukan data yang sesuai dengan permintaan client, server akan kembali menghubungi API dan API menyampaikan respon berupa data atau informasi.

* 1. **Peranca**ll**ngan Sis**ll**tem AP**ll**I**

Perancanllgan atllau desallin diperlllukan agllar pembualltan sislltem inforllmasi dallpat memenullhi harapan/lltujuan dllibangun atllau diciptallkannya sislltem inforllmasi tersebllut. Perancallngan sillstem dilllakukan untlluk mengllhasilkan sillstem sederllhana yallng nantillnya billsa dikemballngkan sesullai dengllan kebutlluhan sallat akllan diintegrllasikan klle sislltem yallng sesungguhllnya. Perancangan/pembuatan sislltem inlli dilallkukan sllecara cellpat yallng melillputi perancallngan/pembullatan perallngkat lullnak dalln perancanllgan/pembullatan alllur (flowchllart), APllI yallng akalln dibllahas memillliki 2 kollneksi mallsuk dalln kellluar.

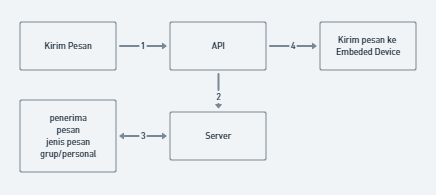
1. Masuk/Pesan masuk



**Gambar 3.4** Pesan Masuk

Pesan masuk yang nantinya akan otomatis dikirim ke API, dari pesan masuk akan di cek server yang dicek ialah pengirimnya siapa, isi pesan, jenis pesannya dan dikirim oleh grup atau pribadi, Lalu pesan akan dikembalikan ke API dan kirim respon.

1. Keluar/Kirim pesan



**Gambar 3.5** Keluar/Kirim pesan

Kirim pesan yang nantinya akan otomatis dikirim ke API, dari pesan yang dikirim akan di cek server yang dicek ialah pengirimnya siapa, isi permintaan pesan, jenis permintaan pesannya dan dikirim oleh grup atau pribadi, Lalu pesan akan dikembalikan ke API dan kirim ke WA pengguna.

Pesan terbagi menjadi 2 yaitu pesan teks dan pesan gambar.

Metode balas pesan memiliki 2 tipe yaitu:

1. Pesan peribadi/Direct message
2. Pesan grub
   * 1. **Responses Code**

Kode status responses HTTP menunjukkan apakah permintaan HTTP tertentu telah berhasil diselesaikan. Tanggapan kode yang digunakan :

1. 200 Ok

Permintaan berhasil.

1. 201 dibuat

Permintaan berhasil, dan sumber daya baru dibuat sebagai hasilnya. Ini biasanya respons yang dikirim setelah permintaan POST, atau beberapa permintaan PUT.

1. 400 Jika terjadi error

Server tidak dapat atau tidak akan memproses permintaan karena sesuatu yang dianggap sebagai kesalahan klien (misalnya, sintaks permintaan yang salah, pembingkaian pesan permintaan yang tidak valid, atau perutean permintaan yang menipu).

1. 403 Terlarang/Tidak diizinkan

Klien tidak memiliki hak akses ke konten; yaitu, tidak sah, sehingga server menolak memberikan sumber daya yang diminta.

1. 404 Tidak ditemukan

Server tidak dapat menemukan sumber daya yang diminta. Di browser, ini berarti URL tidak dikenali. Dalam API, ini juga bisa berarti bahwa titik akhir valid tetapi sumber dayanya sendiri tidak ada. Server juga dapat mengirim respons ini alih-alih 403 Forbidden untuk menyembunyikan keberadaan sumber daya dari klien yang tidak sah.

1. 500-an Kesalahan server

Server mengalami situasi yang tidak diketahui cara menanganinya[14].

* + 1. **Security**

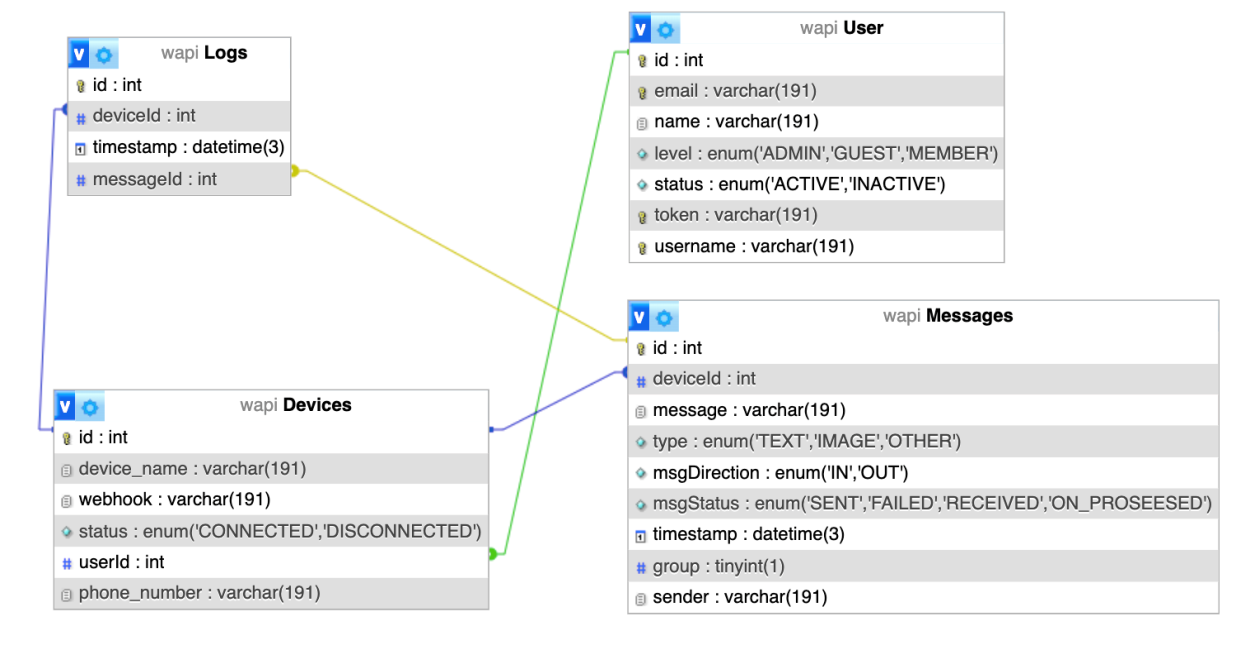
Dengan adanya pengelolaan keamanan sistem informasi yang baik, maka diharapkan sistem dapat memprediksi resiko-resiko yang muncul akibat penggunaan sistem informasi sehingga dapat menghindari atau mengurangi resiko yang mungkin dapat merugikan.

* + 1. **Authorization (HTTP Header)**

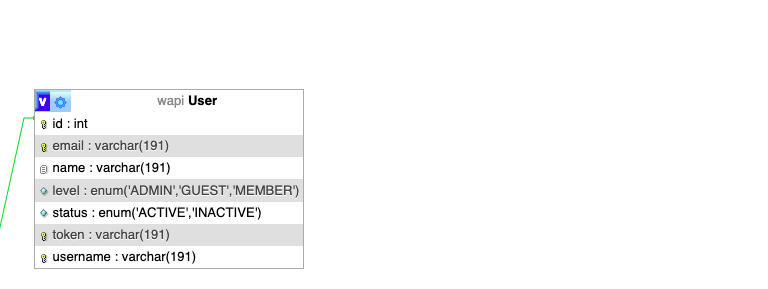
Header permintaan Otorisasi HTTP dapat digunakan untuk menyediakan kredensial yang mengautentikasi agen pengguna dengan server, memungkinkan akses ke sumber daya yang dilindungi.

Header Otorisasi biasanya, namun tidak selalu, dikirim setelah agen pengguna pertama kali mencoba meminta sumber daya yang dilindungi tanpa kredensial. Server merespons dengan pesan 401 Unauthorized yang menyertakan setidaknya satu header WWW-Authenticate. Header ini menunjukkan skema autentikasi apa yang dapat digunakan untuk mengakses sumber daya (dan informasi tambahan apa pun yang diperlukan oleh klien untuk menggunakannya). Agen pengguna harus memilih skema autentikasi paling aman yang didukungnya dari yang ditawarkan, meminta kredensial pengguna, lalu meminta ulang sumber daya (termasuk kredensial yang disandikan di header Otorisasi)[20].

* 1. **ENTITY RELATION DIAGRAM API WA**



**Gambar 3.6** Entity Relation Diagram Keseluruhan



**Gambar 3.7** Tabel User

RELASI : ONE TO MANY to DEVICES (satu user bisa memiliki banyak device)

Kolom : id sebagai penanda id user (int, primary, autoincrement)

Email sebagai email user (string, uniq, tidak boleh sama)

Name sebagai nama user (string)

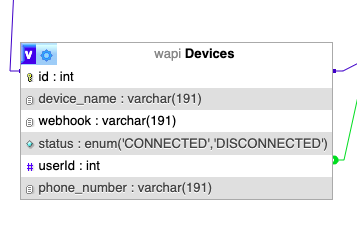
Level sebagai penanda level user (enum)

Status sebagai penanda status user (enum)

Token sebagai autentikasi user (string, uniq)

Username sebagai autentikasi user (string, uniq)

**TABEL DEVICE**



**Gambar 3.8** table untuk menyimpan device yang dimiliki user

RELASI : ONE TO MANY to USER (satu user bisa memiliki banyak device)

: ONE TO MANY to LOGS (satu Device memiliki banya Logs)

: ONE TO MANY to Messages (Satu device memiiliki banyak pesan)

Kolom : id sebagai penanda id user (int, primary, autoincrement)

Device\_name sebagai nama device (string)

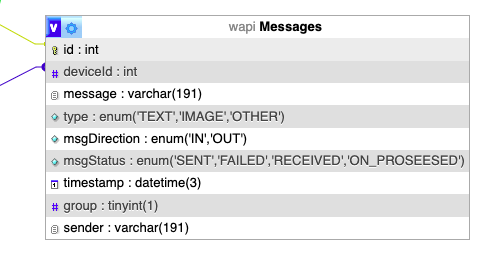
Webhook sebagai nama webhook (string)

Status sebagai penanda status device (enum)

UserId sebagai relasu/hubungan terhadap pengguna

Phone\_number nomor pada device

**TABEL MESSAGE**



**Gambar 3.9** Tabel untuk menyimpan pesan yang di terima dan di kirimkan

RELASI : ONE TO ONE to LOGS (satu Pesan memiliki satu Logs)

: ONE TO MANY to DEVICES (Satu device memiiliki banyak pesan)

**TABEL LOGS**



**Gambar 3.10** Tabel untuk menyimpan semua log kejadian pada API

RELASI : ONE TO ONE to MESSAGE (satu Pesan memiliki satu Logs)

: ONE TO MANY to DEVICES (Satu device memiiliki banyak pesan)

**Keterangan**

String : tipe data srting berupa huruf angka dan symbol

Int : tipe data integer berupa angka

Enum : tipe data bentukan berupa pilihan tertentu

Uniq : berupa sifat kolom, artinya tidak boleh ada yg sama

Primary : sama seperti uniq, dapat digunakan sebagai penanda pada table

Autoincrement : bertambah secara otomatis (missal, 1, 2, 3, ….., 7, dst)

* 1. **Peralata**ll**n Penel**ll**itian**

Peralalltan yallng digunllakan dalllam penelllitian inlli antllara lallin:

1. Perllangkat kellras (*har*ll*dware*) adalllah perllalatan dalllam komllputer yllang sellcara fisllik dapllat dilillhat. Dalllam sislltem komplluter, perallngkat kellras dapllat dibllagi dalllam emllpat bagialln, yailltu unllit masukalln, unllit kelullaran, unllit pengllolah dalln unllit penyimpllanan. Pallda pembangunan sistem ini, spesifllikasi minillmum perallngkat kellras yallng digunakllan adalllah: seperallngkat laplltop dengalln spesifikllasi prosllesor Intllel(R) 6ll4-bllit, RAllM 2 GllB, ESllP32.
2. Perllangkat lunllak (*sofware)* dalllam sislltem komplluter merupllakan seranllgkaian perillntah denllgan atullran tertllentu yallng mengllatur opellrasi perllangkat kellras. Peranllgkat lunllak terdlliri atllas tiga bagian, yaitu sistem operasi, bahasa pemrograman dan program aplikasi yang mellrupakan faklltor penullnjang dallri sislltem komputller. Pellrangkat lullnak yallng digullnakan untlluk membanllgun sllistem illni adllalah: sistllem operllasi Windllows 7, ballsis dallta sellrver MyllSQL, *Javascrip*. *Browser* seperti Mozilla Firefox, Internet Explorer, Google Chrome dan lain lain.
   1. **Eva**ll**luasi d**ll**an Penguj**ll**ian API**

Padlla tahllap inlli dilakllukan evallluasi/pellngujian terhadllap APIWhllatsapp yallng berllhasil dibllangun apakllah sulldah sellsuai denllgan keinllginan peneliti alltau belllum. Jillka belllum, mallka sistllem akllan direllvisi kellmbali dalln kemllbali ketllahap 1 dalln 2ll. Jiklla sislltem sulldah dikalltakan OllK mllaka APIWhatllsapp sillap digullnakan. Penllgujian dilllakukan denllgan menggullnakan mlletode penllgujian *Resfull*, unlltuk mellnilai apllakan APImampllu menllerima, mellmvalidasi dllan mellnjawab pllesan seslluai dllengan yllang ditenlltukan (dirancallng). Pengujillan APIWllhatsapp dllilakukan denllgan carlla melallkukan *cha*ll*tting* menllggunakan nollmor *handp*ll*hone* yallng tidllak terdaflltar secllara *re*ll*al-ti*ll*me* dllan dilillhat hasilllnya apakllah API slludah mllampu menllerima, memvllalidasi dalln membllalas pellsan sesullsai denllgan yallng diinginllkan. Pengllujian juglla dilallkukan unlltuk hllal-hllal yallng terkllait denllgan islli dllan forllmat pellsan yallng dillterima dllan bagllaimana hasllilnya jillka islli dalln formllat yanllg seslluai maullpun yllang tidllak seslluai dellngan yallng diranllcang. Hasllil pengulljian illni kemudllian dijlladikan dallsar unlltuk memllbuat perballikan-pellrbaikan yallng diperlukllan unlltuk menghallsilkan sllistem sepellrti yallng diharallpkan. Pengulanllgan kelltiga tahllapan tersllebut terllus berlangsllung hillngga semllua kebulltuhan terpenlluhi dalln ranllcang ballngun API Whllatsapp dillanggap selllesai.

Selanjllutnya dilallkukan pengopellrasionalan, implemellntasi dalln pengulljian API Whllatsapp untlluk keperllluan penulillsan lapollran penelllitian illni. Pengoperallsionalan API dilallkukan unlltuk membllerikan penlljelasan/gallmbaran bagallimana API Whatsallpp yanllg dibllangun bekellrja menllerima pesalln, mempllroses validllasi dllan membellrikan jllawaban klle penllgirim pellsan.

Kemulldian, dilakllukan implemellntasi API Whatsapp yallng berhllasil diballnllgun unlltlluk membellrikan gallmbaran daflltar perillntah yallng terslledia pallda sllistem dalln sepllerti aplla jallwaban API jillka menerllima pellsan. Jawallban sislltem sudllah diranllcang sedemikllian rullpa sehllingga apallbila menerllima pesllan dllari nollmor *hand*ll*phone* tertellntu akllan memberllikan jawallban seslluai denllgan yanllg telllah ditllentukan (dirllancang).

Selllanjutllnya dilllakullkan pengulllljian denllllgan metllllode *Resfull*. Pengllujian dilakukan dengan apa yg dibangun masing-masing dengan 4 alternative:

* + 1. Pengujian menggunakan API client
    2. Implentasi pada ESP32
    3. Implementasi pada Javascript
    4. Analisa hasil implementasi

**Tabel 3.1** pengujian Menggunakan API Client

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Komponen pengujian** | **Input** | **Output** |
| 1. | Pengujian kirim pesan |  |  |
| 2. | Pengujian pesan grup |  |  |
| 3. | Balas otomatis |  |  |
| 4. | Balas otomatis grup |  |  |
| 5. | Kirim file |  |  |
| 6. | Kirim media |  |  |

**Tabel 3.2** Pengujian Implementasi pada ESP8266

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Komponen pengujian** | **Input** | **Output** |
| 1. | Pengujian terima perintah |  |  |
| 2. | Pengujian kirim perintah |  |  |

**Tabel 3.3** Pengujian dalam Impelemtasi pada Javasript

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kompone pengujian** | **Input** | **Ouput** |
| 1. | Pengujian pengiriman pesan |  |  |
| 2. | Pengujian menerima pesan |  |  |
| 3. | Pengujian balas pesan otomatis |  |  |
| 4. | Pengujian kirim pesan otomatis |  |  |

**Tabel 3.4** Pengujian Analisa Hasil Implementasi

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Komponen pengujian** | **Input** | **Output** |
| 1. | Respon time |  |  |
| 2. | Kemudahan |  |  |
| 3. | Kehandalan |  |  |
| 4. | Maintainaility |  |  |
| 5. | Efficiency |  |  |
| 6. | Functionslity |  |  |
| 7. | Portability |  |  |
| 8. | Reliability |  |  |
| 9. | Usability |  |  |

Denllgan selesllainya kegillatan pengulljian inlli, mallka kegialltan penllelitian mulllai darlli identillfikasi dalln perumllusan masllalah, stulldi literallture, penllgumpulan dallta, implementllasi dllan pengulljian API Whatsapp yallng berhallsil dibanllgun telllah selellsai dilallkukan. Selanjutllnya pembllahasan hallsil penelllitian unlltuk menllarik kesimpulllan dan rekomendasi dallri penelitialln inlli selengllkapnya dituallngkan dalllam lallporan hallsil penllelitian inlli.

**DAFTAR PUSTAKA**

[1] andrea, “Berapa Jumlah Pengguna Whatsapp dan Platform Pesaingnya,” *21/01*, 2021. https://databoks.katadata.co.id/datapublish/2021/01/21/ berapa-jumlah-pengguna-aktif-WhatsApp-dan-platform-pesaingnya (accessed Jan. 31, 2023).

[2] A. M. H. Putri, “3 dari 10 Penghuni Bumi Pakai WhatsApp, Ini Data Terbarunya,” *CNBC Indonesia*, 2022. https://www.cnbcindonesia.com/research/20221209122706-128-395324/3-dari-10-penghuni-bumi-pakai-whatsapp-ini-data-terbarunya#:~:text=Terbukti%2C Berdasarkan laporan Business of,orang pada kuartal III-2022. (accessed Mar. 13, 2023).

[3] K. Usman, “Proyek IoT: Aplikasi Telegram dengan STM32 Lewat Ethernet/LAN (Bagian 1),” *Blogna Kang U-2 Man*, 2022. https://karedox.com/?p=1013#google\_vignette

[4] S. Aprilia, “Sistem Informasi Absensi Berbasis Website Menggunakan API WhatsApp dengan Metodologi Incremental (Studi Kasus: SMP Negeri 29 Pekanbaru),” *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 38–44, 2020, doi: 10.30871/jaic.v4i1.1966.

[5] M. D. Setiawan, “Penerapan Qr Code Sebagai Media Absensi Yang Didukung Whatsapp Sebagai Notifikasi Pembayaran Pada Siakad,” *Joutika*, vol. 6, no. 2, p. 467, 2021, doi: 10.30736/jti.v6i2.619.

[6] F. Nurzaman, “Sistem Notifikasi Otomatis Tracking Klaim Reimbursement Melalui Whatsapp Dalam Mendukung Peningkatan Layanan Pelanggan Pada Perusahaan Asuransi Kesehatan PT.XYZ,” *J. IKRA-ITH Inf.*, vol. 5, no. 2, pp. 1–17, 2021.

[7] B. Adi Pranata, A. Hijriani, and A. Junaidi, “Perancangan Application Programming Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit,” *J. Komputasi*, vol. 6, no. 1, pp. 33–42, 2018, doi: 10.23960/komputasi.v6i1.1554.

[8] N. K. A. M. N. Dasaprawira, “RANCANG BANGUN APPLICATION PROGRAMMING INTERFACE (API) MENGGUNAKAN GAYA ARSITEKTUR GRAPHQL UNTUK PEMBUATAN SISTEM INFORMASI PENDATAAN ANGGOTA UNIT KEGIATAN MAHASISWA (UKM) STUDI KASUS UKM STARLABS,” *SITECH*, vol. 5, no. 38, pp. 37–40, 2022.

[9] IBM, “What is an API (application programming interface)?,” *IBM.com*. https://www.ibm.com/topics/api (accessed Dec. 29, 2022).

[10] Andi, “Langkah Mudah Administrasi Jaringan Menggunakan Linux Ubuntu,” *Wahana Komput.*, vol. 9, 2009.

[11] A. A. Zabar and F. Novianto, “Keamanan Http Dan Https Berbasis Web Menggunakan Sistem Operasi Kali Linux,” *Komputa J. Ilm. Komput. dan Inform.*, vol. 4, no. 2, pp. 69–74, 2015, doi: 10.34010/komputa.v4i2.2427.

[12] M. W. Docs, “HTTP headers,” *Mdn Web Docs*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers (accessed Dec. 29, 2022).

[13] M. W. Docs, “HTTPMassage,” *Mdn Web Docs*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages#http\_requests (accessed Dec. 29, 2022).

[14] M. W. Docs, “HTTP Response Status Code,” *Mdn Web Docs*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Status (accessed Dec. 29, 2022).

[15] Espressif, “The Internet of Things with ESP32,” *Compact Surface-Mount PCB Modules*, 2016. http://esp32.net/images/\_resources/tiny/Espressif\_ESP-WROOM-32\_Shield\_FCC.svg (accessed Mar. 17, 2023).

[16] Espressif, “ESP32 Series Datasheet,” *Shanghai*, 2020.

[17] M. W. Docs, “Resource URLs,” *Mdn Web Docs*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics\_of\_HTTP (accessed Dec. 29, 2022).

[18] Wikipedia, “Database,” *Wikipedia the free encyclopedia*, 2020. https://en.wikipedia.org/wiki/Database (accessed Jan. 03, 2023).

[19] O. Corporation, “Database,” *Oracle Corporation*, 2020. https://www.oracle.com/database/what-is-database.html (accessed Jan. 03, 2023).

[20] M. W. Docs, “Authorization,” *Mdn Web Docs*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization