



KEMENTERIAN
KESEHATAN
REPUBLIK
INDONESIA

Dokumentasi Teknis

SATUSEHAT

Master Patient Index (MPI) ReST API

Pusat Data dan Teknologi Informasi - Digital Transformation Office

Versi 2.1, 23 Oktober 2023
(master-b3a4318, 2023-10-24)

KERAHASIAAN INFORMASI

Informasi yang terkandung dalam dokumen ini bersifat:

PUBLIK

Yang berarti bahwa semua pihak/entitas yang mendapatkan dokumen ini dan/atau sebagian/keseluruhan informasi di dalam dokumen ini, **DIPERBOLEHKAN** untuk menggunakan, menerbitkan, menyebarluaskan, menduplikasi, atau mendistribusikan kembali dokumen ini dan/atau informasi di dalamnya dalam bentuk apapun, baik secara elektronik maupun mekanik tanpa izin tertulis dari **Pusat Data dan Teknologi Informasi – Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Pusat Data dan Teknologi Informasi – Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia tidak membuat pernyataan dan tidak memberikan jaminan dalam bentuk apapun sehubungan dengan informasi di dalam dokumen ini, termasuk tetapi tidak terbatas pada keakuratan atau kelengkapan informasi, fakta dan/atau pendapat yang terkandung di dalam dokumen ini.

Pusat Data dan Teknologi Informasi – Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia, para direktur, para karyawan, dan unit-unit di bawahnya tidak dapat dimintai pertanggungjawaban atas penggunaan dan kepercayaan atas opini, perkiraan, dan temuan dalam dokumen ini.

RIWAYAT PERUBAHAN

Versi	Tanggal Pembaruan	Deskripsi Perubahan	Author/Editor	Versi Aplikasi
2.1	23 Oktober 2023	Update parameter Pencarian Pasien tanpa NIK	Kiky	
2.0	18 Oktober 2023	1. Tambah materi Pasien tanpa NIK 2. Update contoh data <i>Response 2xx</i> 3. Update gambar Pengajuan dan Verifikasi Akses	Kiky	
1.1	05 September 2023	Pemakaian <i>placeholder</i> untuk value <i>client_id</i> , <i>client_secret</i> , <i>access_token</i> dan <i>developer.email</i>	Miko	
1.0	28 Juli 2023	Rilis Dokumentasi Teknis SATUSEHAT: Master Patient Index (MPI) versi AsciiDoc pdf	Ilham	
0.0	27 Juli 2023	Penyusunan Dokumentasi Teknis SATUSEHAT: Master Patient Index (MPI) versi AsciiDoc pdf	Miko	

DAFTAR ISI

KERAHASIAAN INFORMASI	1
RIWAYAT PERUBAHAN	2
PENGANTAR TEKNIS	6
1. Kata Kunci Prioritas	6
2. Simbol dan Istilah	7
3. ReST API	7
3.1. HTTP Method	7
3.2. HTTP Status Codes	8
4. ISO 8601	9
4.1. Format Tanggal dan Waktu	9
5. Pesan Prioritas	10
5.1. Catatan	10
5.2. Tips	10
5.3. Penting	11
5.4. Perhatian	11
5.5. Peringatan	11
A. PENDAHULUAN	12
1. Latar Belakang	12
2. Tujuan	13
3. Sistematika Dokumentasi	13
B. PASIEN DENGAN NIK	15
1. Ruang Lingkup	15
1.1. Persyaratan Data Minimum	15
2. Alur Pembuatan Data	17
3. Alur Mendapatkan Data	17
4. Proses Verifikasi	18
C. PASIEN BAYI BARU LAHIR	20
1. Ruang Lingkup	20
1.1. Persyaratan Data Minimum	20
1.2. Persyaratan Data Minimum - Orang Terkait	22
2. Alur Pembuatan Data	22
3. Alur Mendapatkan Data	25
4. Proses Verifikasi	26
D. PASIEN TANPA NIK	28
1. Ruang Lingkup	28
1.1. Persyaratan Data Minimum	28
2. Alur Mendapatkan Data	28

E. AUTENTIKASI	30
1. Endpoint	30
2. Postman Collection	30
3. Pengajuan dan Verifikasi Akses	31
4. Autentikasi - Mendapatkan Token	37
4.1. Request	37
URL	37
HTTP Verb/Method	37
Header	37
Query String	37
Body (<i>application/x-www-form-urlencoded</i>)	37
4.2. Response	38
2xx Success	38
4xx Client Error	39
5xx Server Error (<i>Content-Type: text/plain</i>)	39
4.3. Contoh Penggunaan/Kode	39
cURL (Windows)	39
cURL (Linux)	39
Postman	40
F. REST API	41
1. Patient - Pencarian Data	42
1.1. Request	42
URL	42
HTTP Verb/Method	42
Header	42
Query String	42
1.2. Response	44
2xx Success	44
4xx Client Error	54
5xx Server Error (<i>Content-Type: text/plain</i>)	54
1.3. Contoh Penggunaan/Kode	54
cURL (Windows)	54
cURL (Linux)	54
Postman	54
2. Patient - Detail Data	55
2.1. Request	55
URL	55
HTTP Verb/Method	55
Parameter Path URI	55
Header	56
2.2. Response	56

2xx Success	56
4xx Client Error	56
5xx Server Error (Content-Type: text/plain)	56
2.3. Contoh Penggunaan/Kode	57
cURL (Windows)	57
cURL (Linux)	57
Postman	57
3. Patient - Penambahan Data	57
3.1. Request	58
URL	58
HTTP Verb/Method	58
Header	58
Body (application/json), dengan identifier NIK Pasien	58
Body (application/json), dengan identifier NIK Ibu	59
3.2. Response	59
2xx Success	59
4xx Client Error	59
5xx Server Error (Content-Type: text/plain)	60
3.3. Contoh Penggunaan/Kode	60
cURL (Windows)	60
cURL (Linux)	60
Postman	60
4. Patient - Pembaruan Sebagian Data	61
4.1. Request	61
URL	61
HTTP Verb/Method	61
Parameter Path URI	61
Header	61
Body (application/json)	62
4.2. Response	63
2xx Success	63
4xx Client Error	63
5xx Server Error (Content-Type: text/plain)	63
4.3. Contoh Penggunaan/Kode	63
cURL (Windows)	63
cURL (Linux)	64
Postman	64
DAFTAR ISTILAH	66
REFERENSI	67

PENGANTAR TEKNIS

Beberapa simbol, istilah, dan konvensi digunakan pada pedoman ini agar penyampaian konsep, struktur data, baik ke tim *developer* maupun non *developer* menjadi lebih jelas. Pada bagian ini akan dijelaskan lebih lanjut terkait hal tersebut.



Setiap teks yang berwarna **biru muda**, dapat diklik untuk melompat ke bagian yang direferensikan.

1. Kata Kunci Prioritas

Dalam banyak dokumen terkait standar, beberapa kata digunakan untuk menandakan persyaratan dalam spesifikasi. Kata-kata ini sering dikapitalisasi dan/atau ditebalkan. Pada bagian ini akan didefinisikan kata-kata ini seperti yang dilakukan pada banyak dokumen RFC (*Request for Comment*), dengan beracuan pada standar di **RFC 2119**[<https://www.ietf.org/rfc/rfc2119.txt>] beberapa kata-kata tersebut dialihbahasakan ke bahasa Indonesia.

- **WAJIB**, berarti definisi tersebut merupakan persyaratan mutlak dari spesifikasi.
- **TIDAK BOLEH**, berarti definisi tersebut merupakan larangan mutlak terhadap spesifikasi.
- **SEBAIKNYA**, berarti bahwa ada alasan yang sah dan kuat dalam keadaan tertentu untuk mengesampingkan hal tertentu lainnya, tetapi implikasi dari perilaku tersebut harus dipahami dan dipertimbangkan dengan cermat sebelum memilih alur yang berbeda.
- **SEBAIKNYA TIDAK**, berarti bahwa ada alasan yang sah dan kuat dalam keadaan tertentu ketika perilaku tertentu dapat diterima atau bahkan berguna, tetapi implikasi dari perilaku harus dipahami dan kasusnya dipertimbangkan dengan cermat sebelum menerapkan perilaku apa pun yang dijelaskan dengan label ini.
- **DIREKOMENDASIKAN**, sama definisinya dengan **SEBAIKNYA**.
- **TIDAK DIREKOMENDASIKAN**, sama definisinya dengan **SEBAIKNYA TIDAK**.
- **BOLEH**, berarti bahwa suatu spesifikasi benar-benar opsional. Satu vendor dapat memilih untuk memasukkan spesifikasi tersebut karena kondisi tertentu membutuhkannya atau karena vendor merasa bahwa itu meningkatkan performa tertentu sementara vendor lain mungkin menghilangkan spesifikasi yang sama. Implementasi yang tidak menyertakan opsi tertentu **PERLU** siap untuk beroperasi dengan implementasi lain yang menyertakan opsi, meskipun mungkin dengan fungsionalitas yang berkurang. Dalam hal yang sama, implementasi yang menyertakan opsi tertentu **PERLU** siap untuk beroperasi dengan implementasi lain yang tidak menyertakan opsi (kecuali, tentu saja, untuk fitur yang disediakan opsi.)
- **OPSIONAL**, sama definisinya dengan **BOLEH**.

Penggunaan kata-kata tadi biasanya untuk panduan dari suatu spesifikasi agar berjalan sesuai yang diharapkan. Bisa juga terkait pertimbangan keamanan, dimana kata-kata tersebut sering digunakan untuk menentukan perilaku dengan implikasi keamanan. Efek pada keamanan karena tidak menerapkan **WAJIB** atau **SEBAIKNYA**, atau melakukan sesuatu yang menurut spesifikasi **TIDAK BOLEH** atau **SEBAIKNYA TIDAK** dilakukan mungkin sangat bijak.

2. Simbol dan Istilah

Berikut beberapa simbol dan istilah yang digunakan dalam dokumentasi ini.

Tabel 1. Penamaan dan Simbol

Penamaan	Simbol/Nilai	Penjelasan
<code>string</code>	<code>"</code> atau <code>'</code>	Representasi dari kumpulan karakter/huruf/symbol.
<code>number</code>	<code>"1"</code> , <code>"-1"</code> , <code>"1.0"</code> , <code>'1'</code> , <code>'-1'</code> , <code>'1.0'</code> ,	Representasi dari kumpulan angka baik dari bilangan bulat atau desimal, dalam bentuk <code>string</code> .
<code>uuid</code>	<code>uuid</code> , <code>uuid1</code> , <code>uuid2</code> , <code>uuid3</code> , <code>uuid4</code>	Merepresentasikan ID universal unik dari suatu entitas, yang terdiri dari 36 karakter (32 karakter heksadesimal dan 4 tanda minus), lebih lengkapnya silakan lihat di https://en.wikipedia.org/wiki/Universally_unique_identifier . Contoh data <code>uuid: 123e4567-e89b-12d3-a456-426614174000</code>

3. ReST API

ReST adalah akronim dari *Representational State Transfer* dan salah satu tipe arsitektur untuk sistem hypermedia terdistribusi. Roy Fielding pertama kali mempresentasikannya pada tahun 2000 dalam disertasinya yang terkenal [\[restfulapi.net\]](http://restfulapi.net) [\[wikipedia-rest\]](http://wikipedia-rest).

ReST melakukan pertukaran data sesuai standar yang ada pada *HTTP Specification*, sehingga perlu dijelaskan secara singkat terkait beberapa *HTTP method* dan *HTTP status code* yang diantaranya digunakan dalam dokumentasi teknis ini.

3.1. HTTP Method

HTTP Method mendefinisikan metode untuk menunjukkan aksi yang sesuai yang akan dilakukan pada *resource* yang telah teridentifikasi. *Resource* sendiri merepresentasikan data yang sudah ada atau data dinamis yang tergantung dari apa yang sudah disepakati dan diterapkan pada *server* yang dituju [\[wikipedia-http-method\]](http://wikipedia-http-method).

Tabel 2. HTTP Method

Method	Penjelasan
<code>GET</code>	Meminta representasi <i>resource</i> dari sumber tertentu untuk di READ oleh peminta.
<code>POST</code>	Meminta server menerima entitas yang terlampir dalam <i>request</i> sesuai struktur yang disepakati sebelumnya dan memprosesnya sebagai proses CREATE .
<code>PUT</code>	Meminta server menerima entitas yang terlampir dalam <i>request</i> sesuai struktur yang disepakati sebelumnya dan memprosesnya sebagai proses UPDATE dari data yang dimaksud.

Method	Penjelasan
PATCH	Meminta server menerima sebagian entitas yang terlampir dalam <i>request</i> sesuai struktur yang disepakati sebelumnya dan memprosesnya sebagai proses PARTIAL UPDATE dari data yang dimaksud.
DELETE	Meminta agar <i>resource</i> yang dimaksud di- DELETE .

3.2. HTTP Status Codes

HTTP Status Codes dikeluarkan dari *server* sebagai bentuk *response* dari *request* yang berasal dari *client* [wikipedia-http-status-code]. Kode-kode ini dikategorikan menjadi 5 kelas yang dibedakan dari digit pertamanya yaitu: **1xx informational response**, **2xx successful**, **3xx redirection**, **4xx client error**, dan **5xx server error**. Pada bagian ini, hanya akan dijelaskan 3 *HTTP Status Codes* yang sering digunakan di ReST yaitu **2xx**, **4xx**, dan **5xx**.

Tabel 3. HTTP Method

Method	Penjelasan
200 OK	Respon standar untuk permintaan yang berhasil.
201 Created	Permintaan berhasil dan <i>resource</i> yang dimaksud telah dibuat. Biasanya akan ada header location yang menunjukkan letak <i>resource</i> yang sudah dibuat.
202 Accepted	Permintaan berhasil dan <i>resource</i> yang dimaksud telah dijadwalkan akan diproses. Biasanya akan ada header location yang menunjukkan letak <i>resource</i> yang akan menunjukkan status atau hasil proses yang sudah atau akan dikerjakan.
204 No Content	Permintaan berhasil dan <i>resource</i> yang dimaksud telah dibuat, tetapi server tidak menyediakan data kembalian/response apapun (pesan di body kosong).
206 Partial Content	Biasanya digunakan dalam proses <i>download</i> data secara partial (<i>byte serving</i> [https://en.wikipedia.org/wiki/Byte_serving]) sesuai rentang yang didefinisikan di- <i>header</i> .
400 Bad Request	Permintaan gagal dikarenakan ada kesalahan di bagian <i>client</i> , seperti struktur data yang tidak sesuai, masukan data yang tidak sah, atau ukuran <i>resource</i> yang dikirim terlalu besar.
401 Unauthorized	Permintaan gagal dikarenakan perlu ada proses autentikasi (<i>authentication</i>) tertentu untuk mengaksesnya atau hasil autentikasi tidak sah.
403 Forbidden	Permintaan gagal dikarenakan perlu ada proses otorisasi (<i>authorization</i>) tertentu untuk mengaksesnya atau otorisasi tidak ditolak.
404 Not Found	Permintaan gagal dikarenakan <i>resource</i> yang dimaksud tidak ditemukan.
405 Method Not Allowed	Permintaan gagal dikarenakan <i>HTTP Method</i> yang digunakan untuk meminta <i>resource</i> tidak sesuai atau tidak diperbolehkan.

Method	Penjelasan
409 Conflict	Permintaan gagal dikarenakan adanya konflik terkait <i>resource</i> yang diakses, misal pembaruan <i>resource</i> yang sama secara bersamaan.
413 Payload Too Large	Permintaan gagal dikarenakan ukuran <i>payload</i> yang dikirimkan melebihi batas yang bisa diproses oleh <i>server</i> .
415 Unsupported Media Type	Permintaan gagal dikarenakan <i>server</i> tidak mendukung jenis <i>resource</i> yang dikirimkan.
422 Unprocessable Entity	Permintaan gagal dikarenakan adanya galat terkait semantik dari data atau <i>resource</i> yang dikirimkan saat diproses.
429 Too Many Requests	Permintaan gagal dikarenakan <i>client</i> terlalu cepat atau banyak mengakses <i>server</i> , biasanya galat ini terkait dengan ketentuan dari skema <i>rate-limiter</i> .
500 Internal Server Error	Permintaan gagal dikarenakan ada kesalahan dibagian <i>server</i> .
501 Not Implemented	Target permintaan yang dimaksud belum diimplementasikan di <i>server</i> .
503 Service Unavailable	<i>Server</i> dalam kondisi mati atau dalam pemeliharaan (<i>maintenance</i>).
504 Gateway Timeout	<i>Server</i> hanya bertindak sebagai <i>gateway</i> atau <i>proxy</i> dan menerima <i>response</i> yang salah dari <i>server</i> sumber.

4. ISO 8601

ISO 8601 adalah sebuah standar internasional yang digunakan pada pertukaran data terkait tanggal dan waktu [wikipedia-iso8601]. Format tanggal yang digunakan pada dokumentasi teknis ini sebagian besar mengacu pada standar ISO 8601 untuk merepresentasikan tanggal dan waktu.

4.1. Format Tanggal dan Waktu

Tabel 4. Format ISO8601 untuk tanggal dan waktu

Simbol	Penjelasan
YYYY	Merepresentasikan 4 digit tahun, <i>zero-padded</i> dibagian kiri, mulai dari 0000 hingga 9999. Tahun 0000 sama dengan tahun 1 Sebelum Masehi (SM). Representasi tahun Sebelum Masehi harus menggunakan tanda minus -0000 = 1 SM, -0001 = 2 SM, dst.
MM	Merepresentasikan 2 digit bulan, <i>zero-padded</i> dibagian kiri, dalam satu tahun mulai dari 01 hingga 12. Secara berurutan merepresentasikan bulan: 01 Januari, 02 Februari, 03 Maret, 04 April, 05 Mei, 06 Juni, 07 Juli, 08 Agustus, 09 September, 10 Oktober, 11 November, 12 Desember.
DD	Merepresentasikan 2 digit hari, <i>zero-padded</i> dibagian kiri, dalam satu bulan mulai dari 01 hingga 31 tergantung dari bulan yang dimaksud.

Simbol	Penjelasan
hh	Merepresentasikan 2 digit jam, <i>zero-padded</i> dibagian kiri, dari 00 hingga 23.
mm	Merepresentasikan 2 digit menit, <i>zero-padded</i> dibagian kiri, dari 00 hingga 59.
ss	Merepresentasikan 2 digit detik, <i>zero-padded</i> dibagian kiri, dari 00 hingga 60.
.sss	Merepresentasikan 3 digit milidetik, <i>zero-padded</i> dibagian kiri.
.sssss	Merepresentasikan 6 digit nanodetik, <i>zero-padded</i> dibagian kiri.

Contoh penggunaan dari format yang telah dijelaskan sebelumnya:


- YYYY-MM-DD untuk merepresentasikan tanggal, contoh: 2021-01-01 (2 Januari 2021).
- hh:mm:ss untuk merepresentasikan waktu lokal, contoh: 22:10:58 (pukul 22 lewat 10 menit 58 detik).
- YYYY-MM-DD hh:mm:ss untuk merepresentasikan tanggal dan waktu lokal, contoh: 2021-01-02 22:10:58 (2 Januari 2021, pukul 22 lewat 10 menit 58 detik).
- hh:mm:ssZ atau hh:mm:ss+0000 untuk merepresentasikan waktu UTC/GMT, contoh: 10:10:58Z (pukul 10 lewat 10 menit 58 detik UTC) bila dikonversi ke zona Asia/Jakarta (UTC+7) menjadi 17:10:58+0700.
- YYYY-MM-DD hh:mm:ssZ untuk merepresentasikan tanggal dan waktu lokal, contoh: 2021-01-02 10:10:58Z (2 Januari 2021, pukul 10 lewat 10 menit 58 detik UTC) bila dikonversi ke zona Asia/Jakarta (UTC+7) menjadi 2 Januari 2021, 17:10:58+0700.
- YYYY-MM-DD hh:mm:ss.sssss untuk merepresentasikan tanggal dan waktu lokal hingga ke 3 digit mili detik, contoh: 2021-01-02 10:10:58.353 (2 Januari 2021, pukul 10 lewat 10 menit 58 detik 353 mili detik).

5. Pesan Prioritas

Beberapa pesan prioritas berisi pernyataan, peringatan, perhatian, atau teguran tertentu sebagai salah satu cara untuk menekankan beberapa penjelasan yang dianggap penting, digunakan pada dokumentasi ini. Tipe-tipe dari kotak prioritas tersebut antara lain:

5.1. Catatan


Memberikan catatan singkat terkait informasi tertentu. Contoh:



Postman adalah salah satu aplikasi untuk melakukan *testing* terkait ReST API yang *user-friendly* dibandingkan cURL.

5.2. Tips

Memberikan saran kepada pembaca yang terkait *best practices* atau tips tertentu. Contoh:



Untuk versi cURL terbaru, gunakan parameter `--oauth2-bearer` untuk melakukan *request* yang membutuhkan *token bearer*.

5.3. Penting

Memberikan saran kepada pembaca bahwa anjuran yang disebutkan penting untuk dibaca. Contoh:



Selalu pakai *token bearer* yang didapatkan setiap melakukan *request* API yang bukan untuk autentikasi.

5.4. Perhatian

Memberikan saran kepada pembaca agar anjuran yang telah dijelaskan dilakukan dengan hati-hati. Contoh:



Mohon diperhatikan saat melakukan *request* data, dikarenakan ada perbedaan target URL untuk versi *production* dan *development*.

5.5. Peringatan

Memberikan tekanan kepada pembaca agar anjuran yang telah dijelaskan dilakukan dengan serius dan hati-hati, dikarenakan kemungkinan akan ada efek samping. Contoh:



Cek apakah *payload* yang akan dikirimkan sudah sesuai standar yang disepakati sebelumnya, karena bila tidak sesuai *payload* tersebut tidak akan dapat diproses.

A. PENDAHULUAN

1. Latar Belakang

Saat ini untuk mendapatkan informasi detail personal seseorang harus melakukan cek ke API Dukcapil dimana input data berupa NIK lalu outputnya berupa detail informasi personal dari NIK tersebut. Permasalahannya API Dukcapil kini dibatasi hanya pengecekan validitas NIK dan tidak dapat lagi autofill data penduduk by NIK seperti sebelumnya dikarenakan kebijakan perlindungan data dari Dukcapil.

Hal ini membuat sistem yang selama ini tergantung pada *autofill* data NIK menjadi terganggu dan terpaksa melakukan penginputan data penduduk secara manual dan ini berpotensi mengganggu layanan karena proses yang dibutuhkan lebih lama.

Di sisi lain, permasalahan dalam sentralisasi pendataan pasien di Indonesia juga terletak pada belum tersedianya unique identifier untuk pasien yang tidak memiliki NIK. Pasien dalam kategori ini mencakup bayi yang baru lahir, anak dari WNI yang berdomisili di luar negeri, warga negara asing, serta penduduk lainnya yang belum tercatat di Dukcapil.

Sebagai upaya untuk menanggulangi permasalahan ini, sistem pencatatan menggunakan IHS Number, nomor urut yang akan dihasilkan oleh IHS (Indonesia Health Service) untuk setiap *record*, seperti pasien, tenaga kesehatan, fasilitas kesehatan, alat kesehatan, dan lain-lain, telah diinisiasi. Dan disaat yang bersamaan, tengah dirumuskan juga perubahan logika pemberian identifier untuk membedakan pencatatan pasien dengan *records-records* lainnya seperti tenaga kesehatan, fasilitas kesehatan, dan sebagainya, agar dapat saling dibedakan di dalam satu sistem yang sama. IHS Number ini akan menjadi unique identifier yang dapat digunakan system pihak ketiga untuk setiap Master Index di IHS.

Karena kita tidak dapat menggantungkan layanan dari API Dukcapil tersebut, pada saat ini dibutuhkan pengerjaan tambahan untuk membuat database yang berisi kumpulan informasi data personal seseorang yang berasal dari gabungan dari beberapa tabel yang sudah dimiliki tim DTO terkait pelayanan kesehatan. Sistem database ini nantinya juga harus dipastikan dapat menjangkau seluruh pasien yang memiliki ataupun tidak memiliki NIK, sehingga pelayanan kesehatan terhadap seluruh pasien ini dapat berjalan dengan baik.

Sentralisasi data yang berisikan informasi data personal layanan kesehatan ini kita namakan Master Patient Index (MPI) sehingga dapat dimanfaatkan oleh sistem yang terganggu akibat ketergantungan pada API Dukcapil tadi maupun pemangku kepentingan lainnya kedepan. Dalam MPI ini pencatatan data pasien akan menggunakan penomoran yang unik dan standardized dimana nantinya pemangku kepentingan lain dapat menggunakan MPI sebagai tools untuk mendapatkan IHS Number sebagai penomoran utama index pasien.

Apabila melakukan pengiriman data kesehatan melalui **SATUSEHAT** yang memiliki elemen data terkait pasien, maka diperlukan informasi `{patient-ihs-number}` dari pasien yang bersangkutan. `{patient-ihs-number}` seorang pasien didapatkan dari **Master Patient Index (MPI) Kementerian Kesehatan**. MPI menyimpan data-data demografi pasien berskala nasional, mulai dari nama, tanggal lahir, alamat, Identitas resmi yang diterbitkan pemerintah, dan lain lain. Setelah mendapatkan `{patient-ihs-number}`, ID dapat disimpan secara di masing-masing sistem internal fasyankes maupun partner non-fasyankes.

{patient-ihs-number} akan mempermudah pelaporan pelayanan kesehatan yang berhubungan dengan pasien, karena partner tidak diwajibkan menyertakan data diri setiap ada pengiriman data {patient-ihs-number} juga dapat digunakan untuk melihat data diri pasien secara menyeluruh.

Master Patient Index (MPI) merupakan the single-source-of-truth informasi pasien yang dapat diakses oleh fasyankes yang memiliki akses ke IHS. MPI dibagi menjadi beberapa *case* (kasus) yang nantinya akan dibahas lebih dalam pada **Dokumentasi SATUSEHAT Master Patient Index (MPI)** ini. Berikut ini *case* interoperabilitas MPI:

- Pasien dengan NIK.
- Pasien bayi baru lahir.
- Pasien tanpa NIK (contoh WNA).
- Pasien tanpa identitas atau dalam gangguan jiwa.

Disamping adanya dokumentasi secara umum terkait Master Patient Index (MPI), maka diperlukan satu dokumentasi khusus yang membahas semua informasi teknis yang disediakan oleh Master Patient Index (MPI) yang dikembangkan oleh tim *developer Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia*. Dokumentasi Teknis Master Patient Index (MPI) ini berisi spesifikasi teknis dan ReST API mencakup parameter yang tersedia, contoh *request*, contoh hasil *response*, penjelasan dari kode status HTTP yang mungkin diterima, serta cara penggunaannya menggunakan cURL dan Postman.

2. Tujuan

Dokumentasi **Teknis Master Patient Index (MPI)** dibuat untuk kebutuhan informasi teknis/ReST API yang tersedia dari Master Patient Index (MPI) yang telah dikembangkan oleh tim *developer SATUSEHAT*. Penjelasan teknis/ReST API Master Patient Index (MPI) didokumentasi ini hanya terbatas dari spesifikasi dari ReST API itu sendiri, penjelasan lengkap terkait Master Patient Index (MPI), akan dijelaskan pada dokumentasi tersendiri.

Dokumentasi SATUSEHAT Master Patient Index (MPI) ini terbagi menjadi beberapa bagian yang terdiri dari:

1. Pengenalan MPI, berisi penjelasan secara umum terkait;
2. Pasien dengan NIK;
3. Pasien Bayi Baru Lahir;
4. Pasien DGJ (Dalam Gangguan Jiwa);
5. Pasien Tanpa NIK;
6. Autentikasi;
7. ReST APIMaster Patient Index (MPI).

3. Sistematika Dokumentasi

Sistematika pembahasan ReST API Master Patient Index (MPI) pada dokumentasi ini dibagi menjadi beberapa bagian yang terdiri dari:

Lembar Kerahasiaan Informasi, berisi pernyataan terkait siapa saja yang boleh memiliki akses terkait dokumentasi ini.

Lembar Riwayat Perubahan, berisi riwayat perubahan terkait isi dari dokumentasi ini dan detail singkat dari perubahan tersebut.

Daftar Isi, berisi daftar nama bab, sub bab, dan halaman terkait pada dokumentasi teknis ReST API Master Patient Index (MPI) ini.

Pengantar Teknis, berisi pengantar teknis singkat terkait beberapa konvensi yang digunakan pada dokumentasi teknis ini. Di bagian ini juga terdapat penjelasan singkat mengenai ReST API serta format tanggal dan waktu sesuai ISO 8601.

Pendahuluan, berisi penjelasan mengenai latar belakang, tujuan, dan sistematika penyusunan dokumentasi teknis terkait Master Patient Index (MPI) ini.

Autentikasi, berisi penjelasan dan spesifikasi teknis terkait ReST API yang digunakan untuk proses autentikasi ke Master Patient Index (MPI) melalui ReST API yang disediakan.

Daftar Istilah, berisi istilah-istilah yang mungkin ada dan perlu penjelasan lebih lanjut yang sebagian atau selalu digunakan pada dokumentasi teknis ReST API Master Patient Index (MPI) ini.

Referensi, berisi daftar referensi yang digunakan atau untuk penjelasan lebih lanjut dari suatu topik yang dibahas pada dokumentasi teknis ReST API Master Patient Index (MPI) ini.

B. PASIEN DENGAN NIK

1. Ruang Lingkup

Pasien dengan NIK akan memiliki proses sebagai berikut:

- Create Patient Data.
- Get Patient Data.

1.1. Persyaratan Data Minimum

Berikut ini persyaratan data minimum untuk pasien baru dengan NIK:



Setiap terdapat simbol asterik * sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

Tabel 5. Persyaratan data minimum pasien baru dengan NIK

Data Point	Keterangan
*identifier	Identitas yang digunakan oleh Pasien, saat ini Identitas yang dapat digunakan yaitu: NIK (nik)
*name	Nama lengkap pasien sesuai KTP/NIK yang diberikan di fasyankes.
*birthDate	Tanggal Lahir dari Pasien sesuai KTP/NIKi. dengan format: YYYY-MM-DD
birthPlace	Digunakan untuk menyimpan informasi Tempat lahir dari Pasien, sesuai standar penulisan NIK/KTP
*gender	Jenis Kelamin dari Pasien, terdiri dari male , dan female
Nomor Kartu Keluarga	Nomor administrasi keluarga yang diterbitkan DUKCAPIL

Data Point	Keterangan
*address	<p>Alamat pasien sesuai dengan KTP/NIK, informasi alamat terdiri dari komponen:</p> <ul style="list-style-type: none"> • use - peruntukan alamat dalam hal ini menggunakan rumah atau home • line - alamat lengkap sesuai KTP/NIK • city - nama kota • postalCode - kode pos • country - negara, dalam hal ini menggunakan ID • extension - penerapan kode administratif wilayah
*address-extension	<p>Digunakan untuk memuat informasi kode administratif wilayah dari alamat pasien sesuai NIK/KTP, yang terdiri dari:</p> <ul style="list-style-type: none"> • province - Kode Provinsi • city - Kode Kab/Kota • district - Kode Kecamatan • village - Kode Desa/Kelurahan • rt - Nomor RT • rw - Nomor RW <p>Kode administratif mengacu ke kode wilayah kemendagri. Daftar kode wilayah terbaru dapat didapatkan di tautan berikut Kode Wilayah KEMENDAGRI [https://bit.ly/kodewilayahKEMENDAGRI]</p>
telecom	Digunakan untuk menyimpan informasi kontak pasien yang dapat dihubungi seperti telepon atau e-mail
maritalStatus	<p>Digunakan untuk menyimpan informasi status pernikahan dari Pasien, seperti:</p> <ul style="list-style-type: none"> • S - Never Married • M - Married • W - Widowed • D - Divorced
citizenshipStatus	Digunakan untuk menyimpan informasi status kewarganegaraan, dalam hal ini WNI atau WNA



Setiap data harus unik / **TIDAK BOLEH** terdapat duplikasi

2. Alur Pembuatan Data

Untuk membuat data pasien baru dengan NIK di SATUSEHAT, silahkan ikut langkah-langkah berikut:

Tabel 6. Alur pembuatan data (NIK)

Langkah	Approach
Langkah 1: GET <i>Patient Record</i> dari Pasien berdasarkan NIK (Pastikan data pasien ada di FHIR SATUSEHAT)	<ul style="list-style-type: none"> • Search by nomor IHS atau • Search by NIK untuk GET nomor IHS • Parameter yang didapatkan: <ul style="list-style-type: none"> ◦ Nomor IHS
Langkah 1: POST <i>Create Patient record</i> dari Pasien yang memiliki NIK ke APIGee	<ul style="list-style-type: none"> • Parameter ini diperlukan untuk <i>create patient record</i>: <ul style="list-style-type: none"> ◦ NIK sebagai Identifier ◦ Nama ◦ Tanggal Lahir ◦ Tempat Lahir ◦ Jenis Kelamin ◦ Alamat ◦ Provinsi ◦ Kota ◦ Daerah ◦ Kecamatan • Data akan dikirimkan ke API MPI • Periksa kesamaan NIK di MPI • Jika tidak ditemukan maka: <ul style="list-style-type: none"> ◦ Validasi NIK oleh pasien itu sendiri ◦ Validasi NIK akan dilakukan oleh pasien itu sendiri dengan data DUKCAPIL ◦ Jika NIK Valid (sesuai dengan data DUKCAPIL), maka <i>generate</i> Nomor IHS Baru <ul style="list-style-type: none"> ▪ Jika tidak, <i>return Error not Valid</i> DUKCAPIL ke APIGee • <i>Return</i> Nomor IHS ke APIGee

3. Alur Mendapatkan Data

Fasyankes dapat memperoleh data pasien baru dengan NIK di SATUSEHAT, dengan menggunakan opsi berikut:

Tabel 7. Alur mendapatkan data

Options	Approach
Option 1: Gunakan Nomor IHS Pasien ke SATUSEHAT. (Use Patient IHS Number to SATUSEHAT)	<ul style="list-style-type: none"> Jika Nomor IHS sudah benar, maka akan dikirimkan data Pasien. Informasi yang akan dikirim kembali ke APIGEE: <ul style="list-style-type: none"> NIK
Option 2a: Gunakan NIK Pasien sebagai parameter Identifier untuk SATUSEHAT.	<ul style="list-style-type: none"> Memeriksa NIK Informasi yang akan dikirim kembali ke APIGEE: <ul style="list-style-type: none"> IHS Number Pasien Nama Pasien
Option 2b: Gunakan NIK Pasien sebagai parameter Identifier untuk SATUSEHAT dan informasi sesuai KTP/NIK.	<ul style="list-style-type: none"> Memeriksa NIK, Nama, dan Tanggal Lahir Pasien. Informasi yang akan dikirim kembali ke APIGEE: <ul style="list-style-type: none"> Nama Pasien Tanggal Lahir Pasien Tempat Lahir Pasien Alamat Pasien Jenis Kelamin Pasien

4. Proses Verifikasi

Ada beberapa data NIK yang perlu diverifikasi oleh DUKCAPIL sebagai single source of truth (SSOT)

Tabel 8. NIK yang diverifikasi oleh DUKCAPIL

Options	Process
NIK Pasien	<ul style="list-style-type: none"> Setiap data Pasien Baru dibuat atau informasi NIK Baru diperbarui. Data yang akan dikirim untuk verifikasi: <ul style="list-style-type: none"> NIK Nama Lengkap Tanggal Lahir Jenis Kelamin Data Pasien akan diverifikasi ke DUKCAPIL. Jika data <i>record</i> pasien telah terverifikasi maka data tersebut akan diperlakukan sebagai "Golden Record". Setiap data pasien yang terhubung yang dibuat menggunakan Identifier lain akan dirujuk ke "Golden Record".



Untuk Identifier lain, tidak akan ada proses verifikasi, dan akan menggunakan proses

penggabungan untuk menautkan lebih dari satu profil pasien yang cocok.

C. PASIEN BAYI BARU LAHIR

1. Ruang Lingkup

Pasien bayi baru lahir akan memiliki proses sebagai berikut:

- Create Patient Data.
- Get Patient Data.

1.1. Persyaratan Data Minimum

Berikut ini persyaratan data minimum untuk pasien bayi baru lahir:



Setiap terdapat simbol asterik * sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

Tabel 9. Persyaratan data minimum pasien bayi baru lahir

Data Point	Keterangan
*identifier	Identitas yang digunakan oleh Pasien, saat ini Identitas yang dapat digunakan hanya ada 2 (dua) jenis yaitu: NIK (nik) dan NIK Ibu (nik-ibu)
*name	Nama pasien bayi baru lahir yang diberikan di fasyankes. Penamaan yang mungkin: <ul style="list-style-type: none">• Nama lengkap diberikan oleh keluarga• Penamaan generik dari fasyankes, contoh "Bayi Ny. [Nama ibu]"
*birthDate	Tanggal Lahir dari Pasien Bayi. dengan format: YYYY-MM-DD
birthPlace	Digunakan untuk menyimpan informasi Tempat lahir dari Pasien, sesuai standar penulisan NIK/KTP
*gender	Jenis Kelamin dari Pasien, terdiri dari " male ", dan " female "
Nomor Kartu Keluarga	Nomor administrasi keluarga yang diterbitkan DUKCAPIL

Data Point	Keterangan
*multipleBirthInteger	<p>Digunakan untuk menyimpan informasi urutan kelahiran untuk pasien Bayi baru lahir, berisi nominal "1", "2", dan seterusnya sesuai dengan urutan kelahiran dalam kasus kelahiran kembar.</p> <p>Untuk kasus kelahiran tidak kembar dapat memasukkan nilai "0"</p>
*address	<p>Alamat pasien sesuai dengan KTP/NIK, untuk pasien Bayi baru lahir dapat menggunakan informasi alamat dari Orang Tua Bayi, informasi alamat terdiri dari komponen:</p> <ul style="list-style-type: none"> "use" - peruntukan alamat dalam hal ini menggunakan rumah atau "home" "line" - alamat lengkap sesuai KTP/NIK "city" - nama kota "postalCode" - kode pos "country" - negara, dalam hal ini menggunakan "ID" "extension" - penerapan kode administratif wilayah <p>Kode administratif mangacu ke kode wilayah kemendagri. Daftar kode wilayah terbaru dapat didapatkan di tahun berikut Kode Wilayah KEMENDAGRI [https://bit.ly/kodewilayahKEMENDAGRI]</p>
*address-extension	<p>Digunakan untuk memuat informasi kode administratif wilayah dari alamat pasien sesuai NIK/KTP, yang terdiri dari:</p> <ul style="list-style-type: none"> "province" - Kode Provinsi "city" - Kode Kab/Kota "district" - Kode Kecamatan "village" - Kode Desa/Kelurahan "rt" - Nomor RT "rw" - Nomor RW
telecom	Digunakan untuk menyimpan informasi kontak pasien yang dapat dihubungi seperti telepon atau email
maritalStatus	<p>Digunakan untuk menyimpan informasi status pernikahan dari Pasie, seperti:</p> <ul style="list-style-type: none"> S - "Never Married"

Data Point	Keterangan
citizenshipStatus	Digunakan untuk menyimpan informasi status kewarganegaraan, dalam hal ini "WNI" atau "WNA"



Setiap data harus unik / **TIDAK BOLEH** terdapat duplikasi

1.2. Persyaratan Data Minimum - Orang Terkait

Berikut ini persyaratan data minimum untuk orang yang berkaitan dengan bayi baru lahir (Ibu/ Ayah):



Setiap terdapat simbol asterik * sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: *variabel.

Tabel 10. Persyaratan data minimum orang terkait

Data Points	Rationale
*ID Related Person	An unique identifier dari record orang yang terkait (Ibu/ Ayah) - Generate UUID dari FHIR
*Related person's Identifier	Identitas pasien dari orang yang terkait (Ibu/ Ayah) (Nomor IHS)
*Patient's Identifier	Identifier pasien (Nomor IHS)
*Relationship type	Hubungan pasien dengan orang yang terkait (Ibu/ Ayah)
*Name	Nama lengkap orang yang terkait (Ibu/ Ayah)
*Gender	Jenis kelamin orang yang terkait (Ibu/ Ayah)
*Birth Date	Tanggal lahir orang yang terkait (Ibu/ Ayah)
*Address	Alamat orang yang terkait (Ibu/ Ayah)
*Phone Number	Nomor telepon orang yang terkait (Ibu/ Ayah)



Orang terkait mengacu pada pasien terkait, misalnya Ibu, atau Ayah dari bayi yang baru lahir.

2. Alur Pembuatan Data

Untuk membuat data pasien bayi baru lahir di SATUSEHAT, silahkan ikut langkah-langkah berikut:

Tabel 11. Alur pembuatan data (NIK Ibu)

Langkah	Approach
<p>Langkah 1: GET informasi Ibu pasien dari SATUSEHAT</p> <p>(Untuk memastikan data Pasien Ibu sudah ada di SATUSEHAT)</p>	<ul style="list-style-type: none"> • Search by nomor IHS atau • Search by NIK untuk GET nomor IHS • Parameter yang didapatkan: <ul style="list-style-type: none"> ◦ Nomor IHS
<p>Langkah 1.a: POST Create Patient record dari Ibu ke APIGee</p> <p>(Jika record Ibu pasien tidak ditemukan)</p>	<ul style="list-style-type: none"> • Parameter ini diperlukan untuk <i>create patient record</i>: <ul style="list-style-type: none"> ◦ NIK sebagai Identifier ◦ Nama ◦ Tanggal Lahir ◦ Tempat Lahir ◦ Jenis Kelamin ◦ Alamat ◦ Provinsi ◦ Kota ◦ Daerah ◦ Kecamatan • Data akan dikirimkan ke API MPI • Periksa kesamaan NIK di MPI • Jika tidak ditemukan maka: <ul style="list-style-type: none"> ◦ Validasi NIK oleh pasien itu sendiri ◦ Validasi NIK akan dilakukan oleh pasien itu sendiri dengan data DUKCAPIL ◦ Jika NIK Valid (sesuai dengan data DUKCAPIL), maka <i>generate Nomor IHS</i> Baru <ul style="list-style-type: none"> ▪ Jika tidak, <i>return Error not Valid</i> DUKCAPIL ke APIGee • <i>Return</i> Nomor IHS ke APIGee

Langkah	Approach
Step 2.a: POST Informasi Pasien Bayi Baru Lahir ke APIGee	<ul style="list-style-type: none"> Parameter ini diperlukan untuk membuat <i>record</i> pasien bayi baru lahir <ul style="list-style-type: none"> NIK Ibu sebagai Identifier Nama Tanggal Lahir Tempat Lahir* Jenis Kelamin Urutan Kelahiran Alamat (gunakan Alamat Ibu) Provinsi Kota Daerah Kecamatan (Desa) Nomor KK (Opsional) Mengirimkan Data ke MPI API Cek kesamaan pada NIK Ibu untuk duplikasi data berdasarkan parameter: <ul style="list-style-type: none"> Nama Tanggal Lahir Jenis Kelamin Urutan Kelahiran Jika tidak ditemukan duplikasi, maka: Buat Nomor IHS untuk Pasien Bayi Baru Lahir di MPI <ul style="list-style-type: none"> Jika tidak, <i>return</i> Nomor IHS yang ada <i>Return</i> Nomor IHS ke APIGee POST Data Pasien ke FHIR SATUSEHAT dari MPI Lanjutkan ke langkah 2.b untuk membuat <i>resources</i> RelatedPerson

Langkah	Approach
Step 2.b: POST Create record RelatedPerson ke APIGee dan menghubungkan ke <i>record</i> Ibu dari Bayi Baru Lahir (saat ini dijalankan otomatis dari system)	<ul style="list-style-type: none"> Parameter ini diperlukan untuk membuat record RelatedPerson: <ul style="list-style-type: none"> Nomor IHS Ibu Nomor IHS Pasien Relation Type: MTH Nama Ibu Tanggal Lahir Ibu Jenis Kelamin Ibu Alamat Ibu Nomor Telepon Ibu POST RelatedPerson resource record ke FHIR SATUSEHAT dari MPI API service <ul style="list-style-type: none"> Generate RelatedPerson ID PUT record RelatedPerson sebagai LINK ke record Ibu Pasien SAVE RelatedPerson resource record ke MPI Database (MongoDB) Return Success to APIGee

3. Alur Mendapatkan Data

Fasyankes dapat memperoleh data pasien bayi baru lahir di SATUSEHAT, dengan menggunakan opsi berikut:

Tabel 12. Alur mendapatkan data

Options	Approach
Option 1: Gunakan Nomor IHS Pasien ke SATUSEHAT. (Use Patient IHS Number to SATUSEHAT)	<ul style="list-style-type: none"> Jika Nomor IHS sudah benar, maka akan dikirimkan data Pasien. Informasi yang akan dikirim kembali ke APIGEE: <ul style="list-style-type: none"> NIK
Option 2a: Gunakan NIK Pasien sebagai parameter Identifier untuk SATUSEHAT.	<ul style="list-style-type: none"> Memeriksa NIK Informasi yang akan dikirim kembali ke APIGEE: <ul style="list-style-type: none"> IHS Number Pasien

Options	Approach
Option 2b: Gunakan NIK Pasien sebagai parameter Identifier untuk SATUSEHAT.	<ul style="list-style-type: none"> Memeriksa NIK, Nama, dan Tanggal Lahir Pasien. Informasi yang akan dikirim kembali ke APIGEE: <ul style="list-style-type: none"> Nama Pasien Tanggal Lahir Pasien Tempat Lahir Pasien Alamat Pasien Jenis Kelamin Pasien
Option 3a: Gunakan NIK Ibu untuk mendapatkan Data Pasien Bayi Baru Lahir.	<ul style="list-style-type: none"> GET dengan menggunakan identifier system NIK Ibu. Jika ditemukan, maka mengembalikan parameter pasien: <ul style="list-style-type: none"> Nomor IHS Nama Urutan Kelahiran Jika kasus kelahiran kembar, <i>return</i> semua pasien yang menggunakan NIK Ibu. Fasyankes perlu memeriksa pasien mana (jika lebih dari satu) yang perlu dirujuk. Gunakan nomor IHS untuk mendapatkan Informasi Pasien.
Option 3b: Gunakan Informasi Ibu Pasien untuk mendapatkan informasi RelatedPerson .	<ul style="list-style-type: none"> Mendapatkan sumber Ibu pasien, dengan menggunakan metode ini: <ul style="list-style-type: none"> Nomor IHS Ibu, atau Kombinasi NIK Ibu, Nama, dan Tanggal Lahir Memeriksa NIK Ibu, Nama, dan Tanggal Lahir. <i>return</i> nomor IHS Pasien yang berasal dari RelatedPersons record dari informasi Ibu yang memiliki tipe relasi “MTH” atau “Ibu”. Informasi yang akan dikirim: <ul style="list-style-type: none"> Nomor IHS Pasien Nama Pasien Urutan Kelahiran Pasien Fasyankes perlu memeriksa pasien mana (jika lebih dari satu) yang perlu dirujuk. Gunakan nomor IHS untuk mendapatkan informasi Pasien.

4. Proses Verifikasi

Ada beberapa data NIK yang perlu diverifikasi oleh DUKCAPIL sebagai single source of truth (SSOT)

Tabel 13. NIK yang diverifikasi oleh DUKCAPIL

Options	Process
Option 1: NIK Pasien.	<ul style="list-style-type: none"> • Setiap data Pasien Baru dibuat atau informasi NIK Baru diperbarui. • Data yang akan dikirim untuk verifikasi: <ul style="list-style-type: none"> ◦ NIK ◦ Nama Lengkap ◦ Tanggal Lahir ◦ Jenis Kelamin • Data Pasien akan diverifikasi ke DUKCAPIL. • Jika data record pasien telah terverifikasi maka data tersebut akan diperlakukan sebagai "Golden Record". • Setiap data pasien yang terhubung yang dibuat menggunakan Identifier lain akan dirujuk ke "Golden Record".
Option 2: NIK Ibu dari Pasien Bayi Baru Lahir.	<ul style="list-style-type: none"> • Saat NIK Ibu dimasukkan sebagai parameter untuk Bayi Baru Lahir. • NIK Ibu akan diperiksa ke data MPI jika sudah ada data pasien tersebut disana. • Jika tidak ada kecocokan pada NIK pasien, maka perlu dikirim ke DUKCAPIL untuk diverifikasi. • Data yang akan dikirim untuk diverifikasi: <ul style="list-style-type: none"> ◦ NIK Ibu ◦ Nama Ibu • Jika NIK dan nama tidak cocok, NIK Ibu tidak dapat digunakan.

D. PASIEN TANPA NIK

1. Ruang Lingkup

Pasien tanpa NIK akan memiliki proses sebagai berikut:

- Get Patient Data.

1.1. Persyaratan Data Minimum

Berikut ini persyaratan data minimum untuk pasien tanpa NIK:



Setiap terdapat simbol asterik * sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: **variabel*.

Tabel 14. Persyaratan data minimum pasien tanpa NIK

Data Point	Keterangan
<i>*name</i>	Berisi nama, baik sebagian atau lengkap, dari pasien yang akan dicari. Contoh: <i>Budi</i>
<i>*birthDate</i>	Berisi tanggal lahir dengan format salah satu dari <i>YYYY</i> , <i>YYYY-MM</i> , atau <i>YYYY-MM-DD</i> . Contoh: <i>1976-01-06</i>
<i>*gender</i>	Berisi jenis kelamin dari pasien, terdiri dari <i>male</i> (laki-laki)/ <i>female</i> (perempuan).



Setiap data harus unik / **TIDAK BOLEH** terdapat duplikasi

2. Alur Mendapatkan Data

Fasyankes dapat memperoleh data pasien tanpa menggunakan NIK di SATUSEHAT, dengan menggunakan opsi berikut:

Tabel 15. Alur mendapatkan data

Options	Approach
Option 1: Gunakan Nama Lengkap , Tanggal Lahir , serta Jenis Kelamin dari Pasien.	Nama Lengkap Pasien: <ul style="list-style-type: none"> Pencarian nama pasien menggunakan <i>similar case</i>. Dimana ketentuan jumlah minimum pencarian berdasarkan karakter adalah TIDAK BOLEH kurang dari tiga karakter.
	Tanggal Lahir Pasien: <ul style="list-style-type: none"> Pencarian dengan tanggal lahir dari pasien sesuai KTP/NIK, dengan format : YYYY-MM-DD dimana bersifat <i>exact/akurat/tepat/persis</i> (TIDAK BOLEH di luar ketentuan).
	Jenis Kelamin Pasien: <ul style="list-style-type: none"> Pencarian dengan jenis kelamin dari pasien, hanya terdiri dari male, dan female dimana bersifat <i>exact/akurat/tepat/persis</i> (TIDAK BOLEH di luar ketentuan)

Tabel 16. Informasi yang akan dikirim kembali ke APIGEE (Response JSON)

Parameter	Tipe Data	Keterangan
identifier	string	Hanya akan menampilkan IHS Number patient.
name	string	Nama pasien.
gender	string	Jenis kelamin pasien: male (laki-laki) / female (perempuan)
birthdate	date	Tanggal lahir pasien. Contoh: 1976-01-06 (YYYY-MM-DD)
multipleBirthInteger	int	Urutan kelahiran, untuk pasien lahir kembar. Jika bukan kelahiran kembar, maka terisi dengan angka 0 .
address	string	Alamat pasien (ditampilkan dengan masking).
maritalStatus	string	Status pasien: Married / unmarried .

E. AUTENTIKASI

Untuk melakukan transaksi data dari Master Patient Index (MPI), perlu dilakukan proses autentikasi terlebih dahulu agar mendapatkan akses yang tersedia. Autentikasi yang digunakan oleh MPI mengikuti standar protokol OAuth 2 dengan tipe pemberian akses (*grant type*) adalah `client_credentials`.

Autentikasi menggunakan *grant type* `client_credentials` adalah proses autentikasi yang dilakukan antara *server to server*, sehingga tidak ada proses registrasi atau *log in* di sini. Autentikasi dengan tipe tersebut hanya memerlukan data berupa `client_id` dan `client_server`, dimana nilai tersebut didapatkan ketika pihak yang ingin menggunakan atau mengakses MSI ini telah melakukan pengajuan, terdaftar, serta mendapatkan persetujuan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Cara Mendapatkan Nilai dari `client_id` dan `client_secret`

Silakan terlebih dahulu melakukan **Pengajuan dan Verifikasi Akses** pada website <https://satusehat.kemkes.go.id/platform> melalui web *browser* Anda. Untuk informasi lebih lanjut dapat dilihat pada [Pengajuan dan Verifikasi Akses](#)



Setiap teks yang berwarna **biru muda**, dapat diklik untuk melompat ke bagian yang direferensikan.

1. Endpoint

Pada bagian ini akan dijelaskan spesifikasi untuk **ReST API SATUSEHAT Terkait Autentikasi**, yang mempunyai tiga *endpoint* berdasarkan jenis lingkungan pengembangannya (*development environment*) yaitu:

- *development* <https://api-satusehat-dev.dto.kemkes.go.id/oauth2/v1>
- *staging* <https://api-satusehat-stg.dto.kemkes.go.id/oauth2/v1>
- *production* <https://api-satusehat.kemkes.go.id/oauth2/v1>



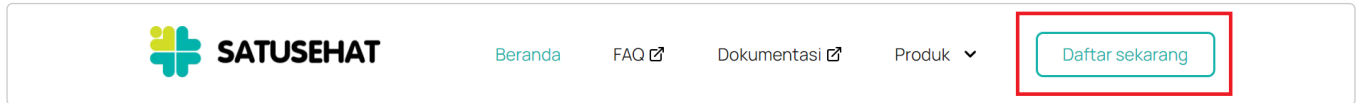
Semua penerapan, penjelasan, dan contoh yang akan dibahas akan menggunakan *endpoint development*.

2. Postman Collection

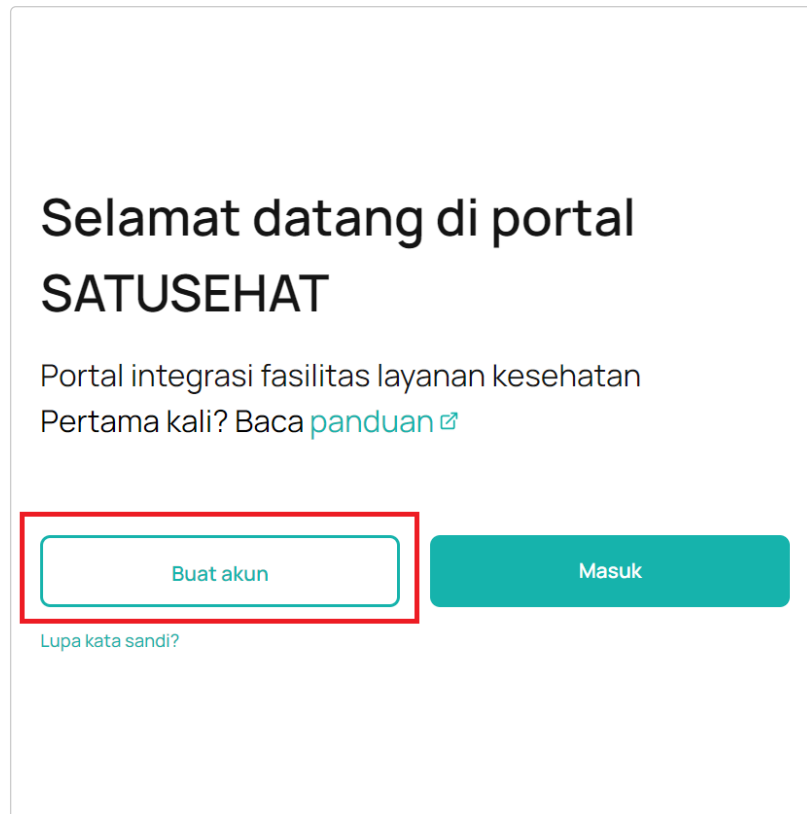
Silakan mengunduh Postman Collection untuk melihat contoh/melakukan workshop secara mandiri pada website [Postman Collection SATUSEHAT](#) [<https://s.id/PostmanSATUSEHAT>] melalui web *browser* Anda.

3. Pengajuan dan Verifikasi Akses

1. Untuk bergabung dengan Ekosistem FASKES SATUSEHAT, akses laman <https://satusehat.kemkes.go.id/platform> melalui web *browser* Anda, kemudian klik **Integrasi Faskes**.



2. Pada laman <https://satusehat.kemkes.go.id/platform/welcome> klik **Integrasi Faskes** untuk membuat akun SATUSEHAT.



3. Lengkapi data identitas berdasarkan *form* yang disediakan, kemudian klik **Buat Akun** untuk mendapatkan konfirmasi aktivasi akun pada *e-mail* yang didaftarkan.

Buat Akun SATUSEHAT Anda

Nama

Email

Nomor telepon

+62

Kata sandi

👁

Minimal 8 karakter termasuk huruf kapital (A-Z), huruf kecil (a-z) dan angka (0-9)

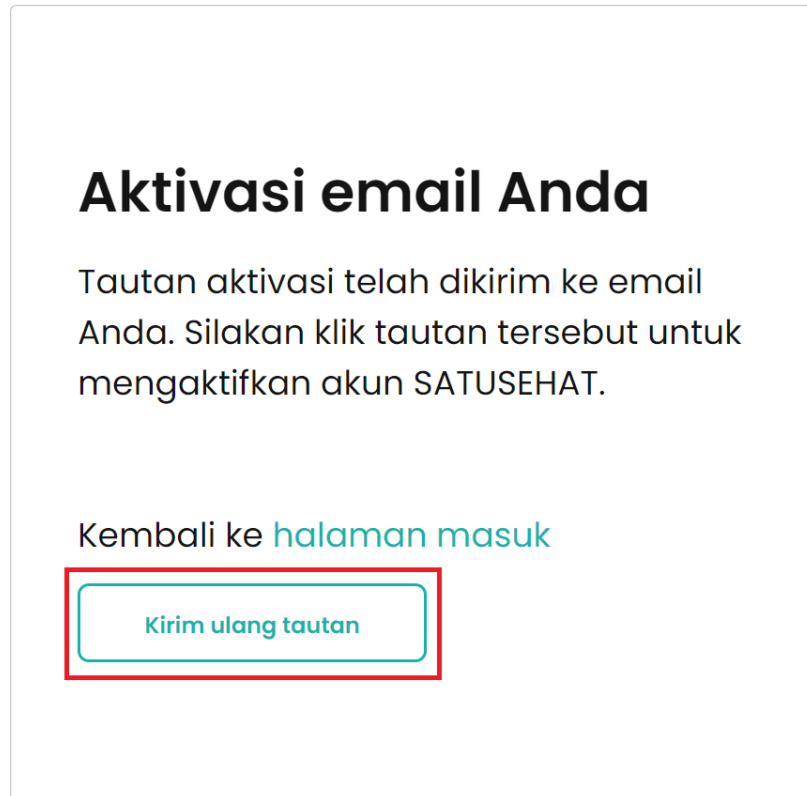
Konfirmasi kata sandi

👁

Minimal 8 karakter termasuk huruf kapital (A-Z), huruf kecil (a-z) dan angka (0-9)

Buat akun

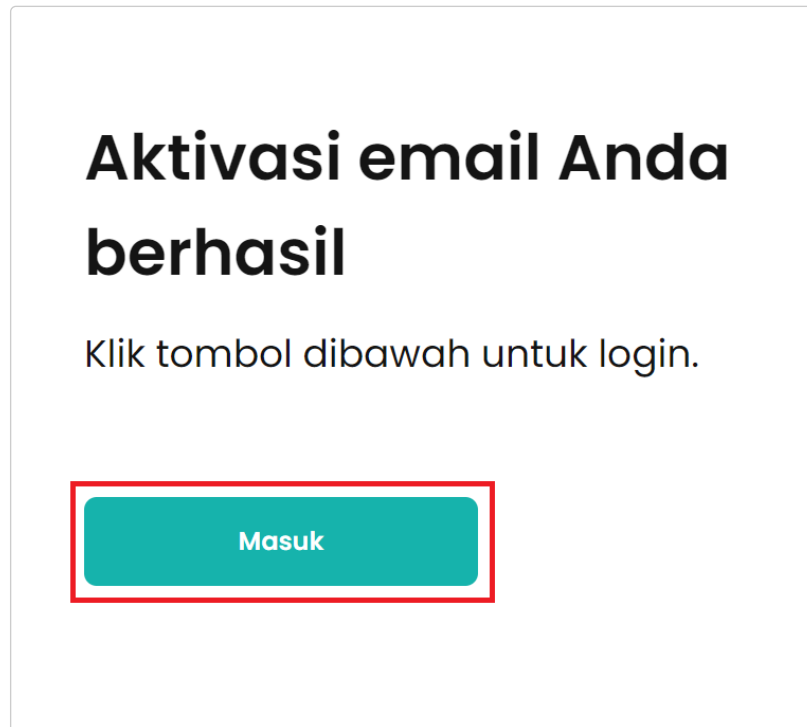
4. Selanjutnya akan muncul laman **Aktivasi e-mail Anda**, apabila aktivasi akun SATUSEHAT tidak terkirim pada *e-mail* yang telah didaftarkan, klik **Kirim ulang tautan**.



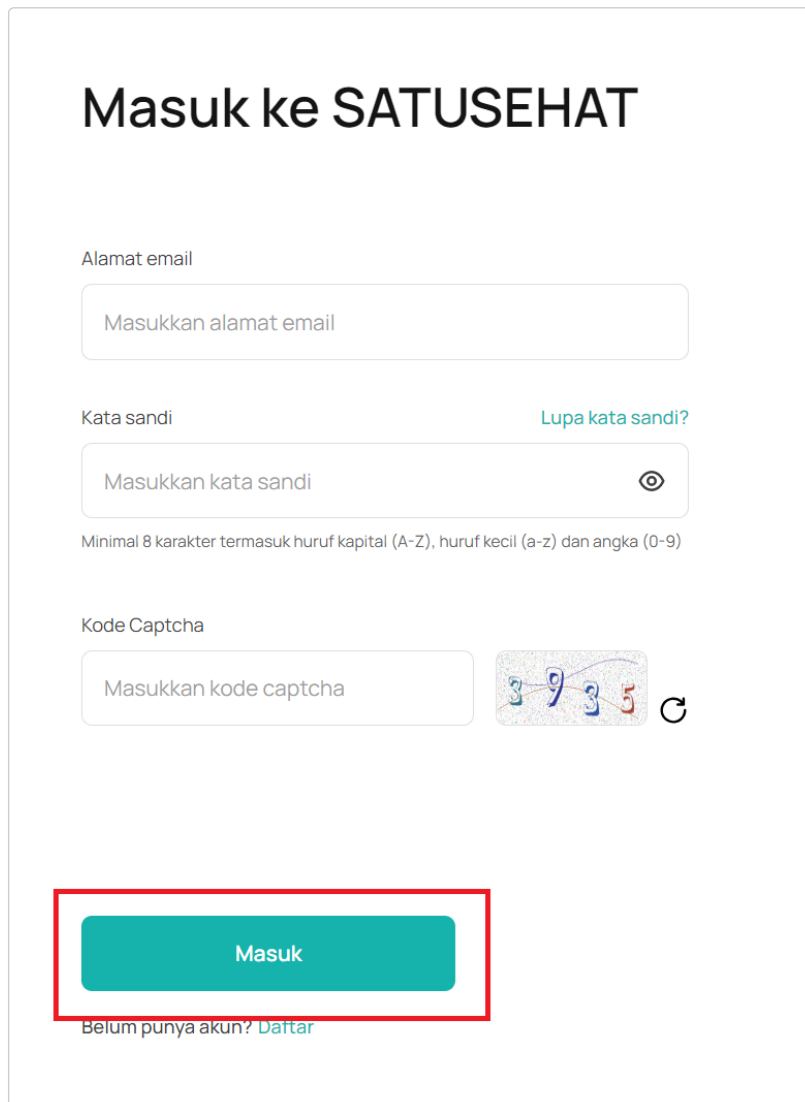
5. Buka *e-mail* aktivasi akun SATUSEHAT, pada *detail*/badan *e-mail* klik **Aktifkan akun** atau dengan menyalin tautan pada browser Anda.



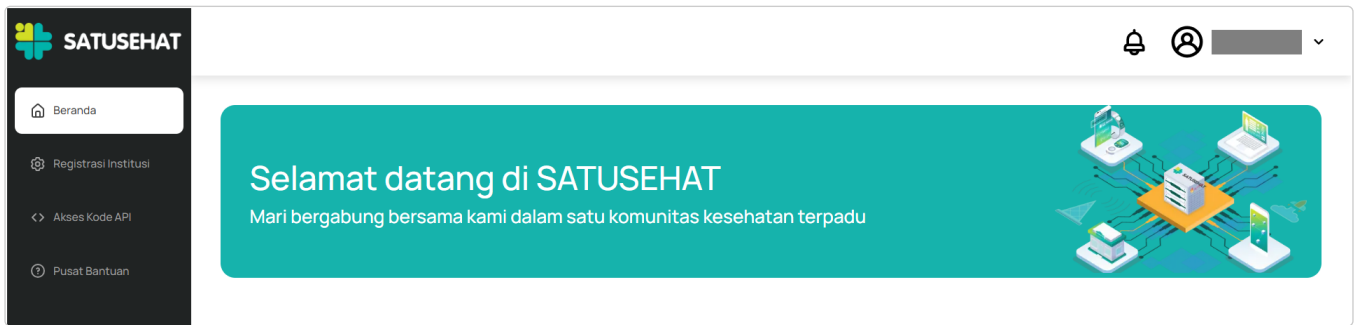
6. Aktivasi akun berhasil, Anda dapat melanjutkan Integrasi Faskes dengan klik **Masuk**.



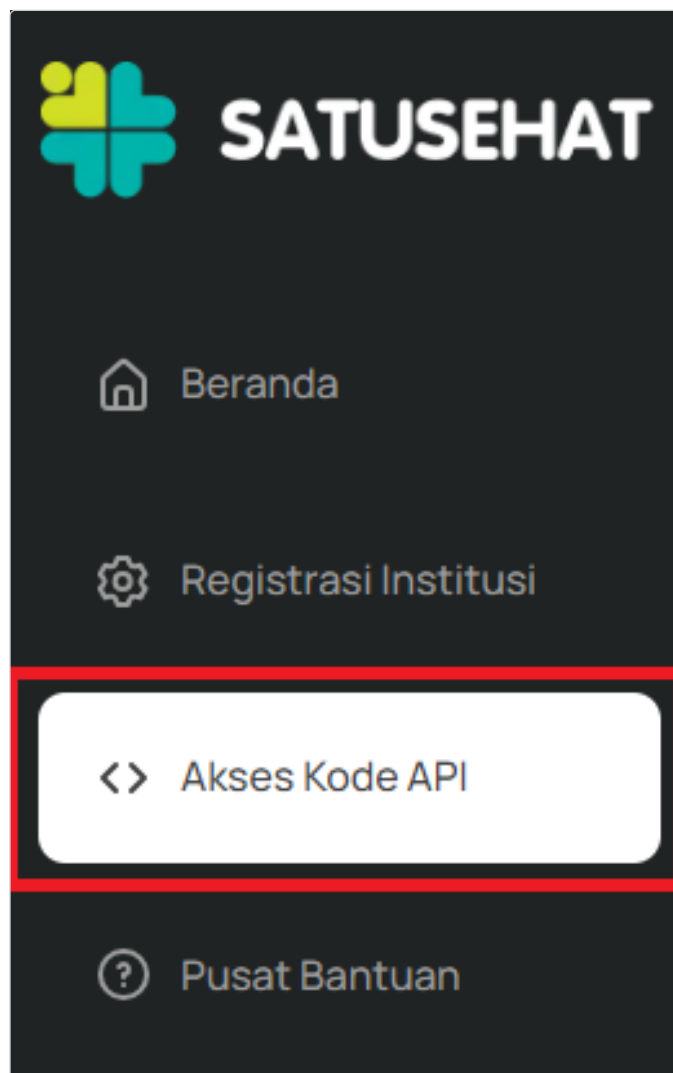
7. Pada laman *login* masukan alamat *e-mail* dan kata sandi yang telah terdaftar, kemudian klik **Masuk**.



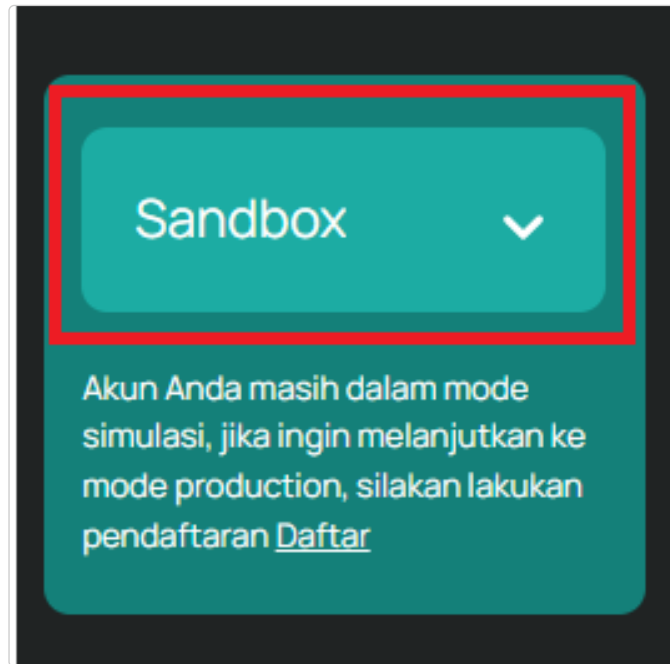
8. Setelah *login*, Anda akan masuk ke laman beranda.



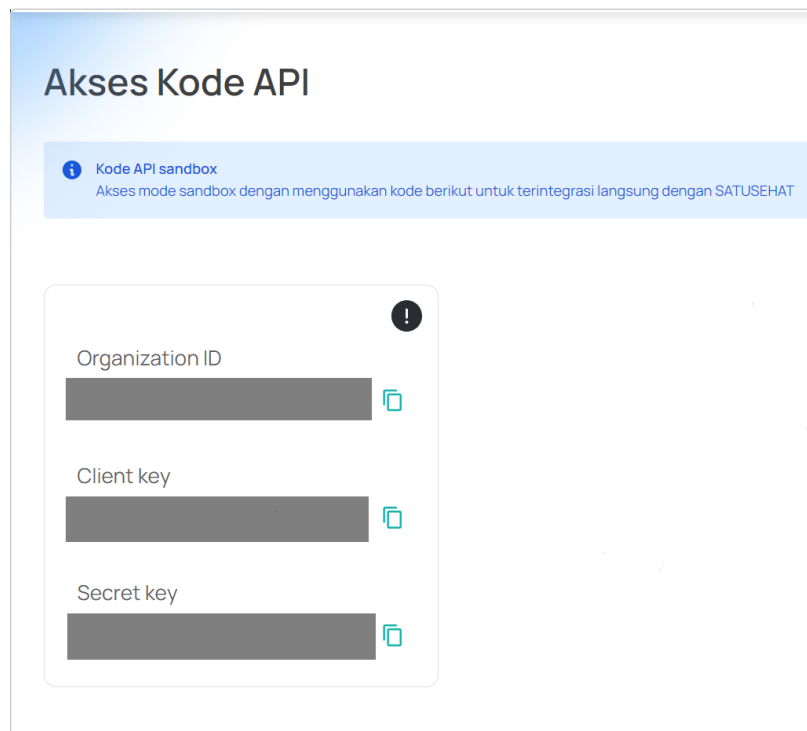
9. Pada *sidebar* menu, klik menu <> **Akses Kode API**.



10. Pada *sidebar* menu, dibawah menu <> **Akses Kode API**. Pilih *endpoint* API yang diinginkan, diantaranya **Sandbox(Development)**, **Staging**, atau **Production**.



11. Anda dapat melihat **Organization ID**, **Client Key**, dan **Secret Key** yang dapat digunakan pada API SATUSEHAT sesuai dengan *endpoint* API yang diinginkan.



4. Autentikasi - Mendapatkan Token

Melakukan proses autentikasi untuk mendapatkan akses token yang akan dipakai pada setiap *request* ReST API Master Patient Index (MPI) (MPI) selanjutnya.



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

4.1. Request

URL

`https://api-satusehat-dev.dto.kemkes.go.id/oauth2/v1/accesstoken`

HTTP Verb/Method

POST

Header

Nama Parameter	Tipe Data	Keterangan
*Content-Type	string	<i>Mime type</i> dari <i>payload</i> data yang akan dikirimkan di dalam <i>body</i> dalam bentuk <i>URL Encoded</i> , WAJIB diisi dengan application/x-www-form-urlencoded

Query String

Nama Parameter	Tipe Data	Keterangan
*grant_type	string	Tipe permintaan akses (<i>grant</i>) Oauth2, WAJIB diisi dengan client_credentials .

Body (**application/x-www-form-urlencoded**)

Nama Parameter	Tipe Data	Keterangan
*client_id	string	Nilai <i>client ID</i> yang telah didapatkan dari Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia setelah melakukan Pengajuan dan Verifikasi Akses pada website https://satusehat.kemkes.go.id/platform , WAJIB diisi. Nilai ini bisa disamakan seperti <i>username</i> yang akan digunakan untuk akses aplikasi.

Nama Parameter	Tipe Data	Keterangan
* <code>client_secret</code>	<code>string</code>	Nilai <i>client secret</i> yang telah didapatkan dari Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia setelah melakukan <i>Pengajuan dan Verifikasi Akses</i> pada website https://satusehat.kemkes.go.id/platform , WAJIB diisi. Nilai ini bisa disamakan seperti kata sandi (<i>password</i>) yang akan digunakan untuk akses aplikasi.

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
client_id: <client-id>
client_secret: <client-secret>
```

4.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola `2xx` atau `4xx`, yang dikembalikan dari server mempunyai parameter *Content-Type* dengan nilai `application/json` di salah satu parameter *header*-nya.

2xx Success

Dari hasil *response* ini, **PERLU** disimpan nilai akses token yang didapat dari properti `access_token`, dimana tipe token (lihat properti `token_type`) tersebut adalah `BearerToken`. Nilai akses token tersebut **WAJIB** selalu digunakan sebagai nilai dari *header* `Authorization: Bearer <access_token>` saat melakukan *request* lainnya dari ReST API SATUSEHAT.

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "refresh_token_expires_in": "0",
  "api_product_list": "[data]",
  "api_product_list_json": [
    "data"
  ],
  "organization_name": "ihs-prod-1",
  "developer_email": "<developer-email>",
  "token_type": "BearerToken",
  "issued_at": "1688958601590",
  "client_id": "<client-id>",
  "access_token": "<access-token>",
  "application_name": "0b1dc094-62cc-4bb1-85a5-616562305ed2",
  "scope": "",
}
```

```
{
  "expires_in": "14399",
  "refresh_count": "0",
  "status": "approved"
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "resourceType": "OperationOutcome",
  //data.terkait.resource.OperationOutcome
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

4.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --insecure --location \
  --header "Content-Type: application/x-www-form-urlencoded" ^
  --data-urlencode "client_id=<client-id>" ^
  --data-urlencode "client_secret=<client-secret>" ^
  --request POST ^
  "https://api-satusehat-dev.dto.kemkes.go.id/oauth2/v1/accesstoken?grant_type=client_credentials"
```


cURL (Linux)

```
curl --insecure --location \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data-urlencode 'client_id=<client-id>' \
  --data-urlencode 'client_secret=<client-secret>' \
  --request POST \
  'https://api-satusehat-dev.dto.kemkes.go.id/oauth2/v1/accesstoken?grant_type=client_credentials'
```


Postman

1. Buat *request* baru menggunakan **New** › **HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-satusehat-dev.dto.kemkes.go.id/oauth2/v1/accesstoken
```

3. Lalu pilih *request method* **POST**.
4. Pada tab **Params**, di bagian **Query Params**:
 - a. masukan nilai **grant_type** pada kotak masukan pada kolom **KEY**,
 - b. lalu masukan nilai **client_credentials** pada kotak masukan pada kolom **VALUE**.
5. Pada tab **Body**:
 - a. pilih **x-www-form-urlencoded**,
 - b. masukan nilai **client_id** pada kotak masukan pada kolom **KEY**,
 - c. lalu masukan nilai *client ID* yang sudah didapatkan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia** pada kotak masukan pada kolom **VALUE**,
 - d. selanjutnya masukan nilai **client_secret** pada kotak masukan pada kolom **KEY**,
 - e. terakhir masukan nilai *client secret* yang sudah didapatkan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia** pada kotak masukan pada kolom **VALUE**.
6. Klik tombol .
7. Hasil *response* akan ditampilkan di bagian **Response**.

F. REST API

Untuk melakukan transaksi data dari Master Patient Index (MPI), perlu dilakukan proses autentikasi terlebih dahulu agar mendapatkan akses yang tersedia. Autentikasi yang digunakan oleh MPI mengikuti standar protokol OAuth 2 dengan tipe pemberian akses (*grant type*) adalah **client_credentials**.

Autentikasi menggunakan *grant type* **client_credentials** adalah proses autentikasi yang dilakukan antara *server to server*, sehingga tidak ada proses registrasi atau *log in* di sini. Autentikasi dengan tipe tersebut hanya memerlukan data berupa **client_id** dan **client_server**, dimana nilai tersebut didapatkan ketika pihak yang ingin menggunakan atau mengakses MSI ini telah melakukan pengajuan, terdaftar, serta mendapatkan persetujuan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Cara Mendapatkan Nilai dari **client_id** dan **client_secret**

Silakan terlebih dahulu melakukan **Pengajuan dan Verifikasi Akses** pada website <https://satusehat.kemkes.go.id/platform> melalui web *browser* Anda. Untuk informasi lebih lanjut dapat dilihat pada [Pengajuan dan Verifikasi Akses](#)



Setiap teks yang berwarna **biru muda**, dapat diklik untuk melompat ke bagian yang direferensikan.

Pada bagian ini akan dijelaskan spesifikasi untuk **ReST API Master Patient Index (MPI)**, yang mempunyai *endpoint* berdasarkan jenis lingkungan pengembangannya (*development environment*) yaitu:

- *development* <https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1>
- *staging* <https://api-satusehat-stg.dto.kemkes.go.id/fhir-r4/v1>
- *production* <https://api-satusehat.kemkes.go.id/fhir-r4/v1>



Semua penerapan, penjelasan, dan contoh yang akan dibahas akan menggunakan *endpoint development*.

1. Patient - Pencarian Data

Fungsi dari ReST API ini adalah untuk mencari data terkait *resource Patient* yang tersedia di *platform SATUSEHAT* dengan parameter-parameter tertentu.



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.



Setiap terdapat simbol tanya **?** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut **WAJIB ada bila memenuhi kondisi tertentu**, contoh: **?variabel**.

1.1. Request

URL

`https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient`

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

1. Pencarian Berdasarkan NIK Pasien

Nama Parameter	Tipe Data	Keterangan
?identifier	string	Parameter ini WAJIB ada bila melakukan pencarian data dengan NIK pasien . Nilai yang dimasukan harus mempunyai format: <div> <code>https://fhir.kemkes.go.id/id/nik <nilai-nik></code> </div> <div>dimana <nilai-nik> berisi NIK dari pasien yang akan dicari. Sebagai contoh, diketahui NIK pasien 3171012345678901, sehingga nilai yang dimasukan untuk parameter ini adalah: <div> <code>https://fhir.kemkes.go.id/id/nik 3171012345678901</code> </div> </div>

2. Pencarian Berdasarkan Nama, Tanggal Lahir, dan NIK Pasien

Nama Parameter	Tipe Data	Keterangan
?name	string	Parameter ini WAJIB ada bila melakukan pencarian data dengan nama, tanggal lahir, dan jenis kelamin pasien. Berisi nama, baik sebagian atau lengkap, dari pasien yang akan dicari. Contoh: smith.
?birthdate	date	Parameter ini WAJIB ada bila melakukan pencarian data dengan nama, tanggal lahir, dan jenis kelamin pasien. Berisi tanggal dengan format salah satu dari YYYY, YYYY-MM, atau YYYY-MM-DD. Terkait penjelasan format tersebut bisa dilihat pada bagian Pengantar Teknis, Format Tanggal dan Waktu . Contoh: 1980-01.
?nik	string	Parameter ini WAJIB ada bila melakukan pencarian data dengan nama, tanggal lahir, dan NIK pasien. Berisi NIK (Nomor Induk Kependudukan) yang terdapat di KTP (Kartu Tanda Penduduk) dari pasien yang bersangkutan. Contoh: 3171012345678901

3. Bayi Baru Lahir - Pencarian Berdasarkan NIK Ibu

Nama Parameter	Tipe Data	Keterangan
?identifier	string	Parameter ini WAJIB ada bila melakukan pencarian data bayi baru lahir dengan NIK ibu. Nilai yang dimasukan harus mempunyai format: <div>https://fhir.kemkes.go.id/id/nik-ibu <nik-ibu></div> dimana <nik-ibu> berisi NIK dari ibu yang akan dicari. Sebagai contoh, diketahui NIK ibu dari pasien 3171012345678901, sehingga nilai yang dimasukan untuk parameter ini adalah: <div>https://fhir.kemkes.go.id/id/nik-ibu 367400001111222</div>

4. Pencarian Berdasarkan Nama, Tanggal Lahir, dan Gender Pasien

Nama Parameter	Tipe Data	Keterangan
?name	string	Parameter ini WAJIB ada bila melakukan pencarian data dengan nama, tanggal lahir, dan jenis kelamin pasien. Berisi nama, baik sebagian atau lengkap, dari pasien yang akan dicari. Contoh: Budi

Nama Parameter	Tipe Data	Keterangan
?birthdate	date	Parameter ini WAJIB ada bila melakukan pencarian data dengan nama, tanggal lahir, dan jenis kelamin pasien. Berisi tanggal dengan format salah satu dari YYYY, YYYY-MM, atau YYYY-MM-DD. Terkait penjelasan format tersebut bisa dilihat pada bagian Pengantar Teknis, Format Tanggal dan Waktu . Contoh: 1976-01-06
?gender	string	Diisi dengan jenis kelamin pasien: male (laki-laki) / female (perempuan) Contoh: male

1.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola 2xx atau 4xx, yang dikembalikan dari server mempunyai parameter *Content-Type* dengan nilai *application/json* di salah satu parameter *header*-nya.

2xx Success

Bila *resource Patient* dengan ID terkait berhasil ditemukan atau tersedia, maka akan mengembalikan data dari *resource Patient* yang tersimpan di *platform SATUSEHAT*.

Contoh Data Pencarian Berdasarkan NIK Pasien

```
{
  "entry": [
    {
      "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02478375538",
      "resource": {
        "active": true,
        "id": "P02478375538",
        "identifier": [
          {
            "system": "https://fhir.kemkes.go.id/id/ihs-number",
            "use": "official",
            "value": "P02478375538"
          },
          {
            "system": "https://fhir.kemkes.go.id/id/nik",
            "use": "official",
            "value": "##### "
          }
        ],
        "link": [
          {
            "other": {
              "reference": "RelatedPerson/7af11905-8d49-47dc-a72b-bca4948c01e7"
            },
            "type": "refer"
          }
        ],
        "meta": {
          "lastUpdated": "2023-05-31T06:40:40.038378+00:00",
          "profile": [
            "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
          ],
          "versionId": "MTY4NTUxNTI0MDAzODM3ODAwMA"
        }
      }
    }
  ]
}
```

```
    },
    "name": [
      {
        "text": "patient 1",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
},
"link": [
  {
    "relation": "search",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-
r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik%7C9271060312000001"
  },
  {
    "relation": "first",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-
r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik%7C9271060312000001"
  },
  {
    "relation": "self",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-
r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik%7C9271060312000001"
  }
],
"resourceType": "Bundle",
"total": 1,
"type": "searchset"
}
```

Contoh Data Pencarian Berdasarkan Nama, Tanggal Lahir, dan NIK Pasien

```
{
  "entry": [
    {
      "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02478375538",
      "resource": {
        "active": true,
        "id": "P02478375538",
        "identifier": [
          {
            "system": "https://fhir.kemkes.go.id/id/ihs-number",
            "use": "official",
            "value": "P02478375538"
          },
          {
            "system": "https://fhir.kemkes.go.id/id/nik",
            "use": "official",
            "value": "#####"
          }
        ],
        "link": [
          {
            "other": {
              "reference": "RelatedPerson/7af11905-8d49-47dc-a72b-bca4948c01e7"
            },
            "type": "refer"
          }
        ],
        "meta": {
          "lastUpdated": "2023-05-31T06:40:40.038378+00:00",
          "profile": [
            "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
          ]
        }
      }
    ]
  },
  "total": 1,
  "type": "searchset"
}
```

```
{
  "versionId": "MTY4NTUxNTI0MDAzODM3ODAwMA",
  "name": [
    {
      "text": "patient 1",
      "use": "official"
    }
  ],
  "resourceType": "Patient"
}
],
"link": [
  {
    "relation": "search",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik%7C9271060312000001"
  },
  {
    "relation": "first",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik%7C9271060312000001"
  },
  {
    "relation": "self",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik%7C9271060312000001"
  }
],
"resourceType": "Bundle",
"total": 1,
"type": "searchset"
}
```

Contoh Data Bayi Baru Lahir - Pencarian Berdasarkan NIK Ibu

```
{
  "entry": [
    {
      "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102825",
      "resource": {
        "active": true,
        "birthDate": "1920-02-28",
        "id": "P02029102825",
        "identifier": [
          {
            "system": "https://fhir.kemkes.go.id/id/nik-ibu",
            "use": "official",
            "value": "367400001111222"
          },
          {
            "system": "https://fhir.kemkes.go.id/id/ihs-number",
            "use": "official",
            "value": "P02029102825"
          }
        ],
        "language": "id",
        "meta": {
          "lastUpdated": "2023-10-17T06:26:21.585017+00:00",
          "profile": [
            "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient|4.0.1",
            "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
          ],
          "versionId": "MTY5NzUyMzk4MTU4NTAxNzAwMA"
        }
      },
      "name": [
        {

```

```
      "text": "Sambo",
      "use": "official"
    }
  ],
  "resourceType": "Patient"
}
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/ee3bff43-6b72-4a72-a41a-af4164e0eacf",
  "resource": {
    "active": true,
    "birthDate": "2023-05-24",
    "id": "ee3bff43-6b72-4a72-a41a-af4164e0eacf",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/paspor",
        "use": "official",
        "value": "A01111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/kk",
        "use": "official",
        "value": "367400001111111"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029103384"
      }
    ],
    "meta": {
      "lastUpdated": "2023-05-25T07:06:42.937195+00:00",
      "profile": [
        "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
      ],
      "versionId": "MTY4NDk5ODQwMjkzNzE5NTAwMA"
    },
    "name": [
      {
        "text": "John Smith",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102814",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102814",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102814"
      }
    ]
  }
}
```



```
    ],
    "meta": {
      "lastUpdated": "2023-03-08T13:18:43.477704+00:00",
      "profile": [
        "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
      ],
      "versionId": "MTY3ODI4MTUyMzQ3NzcwNDAwMA"
    },
    "name": [
      {
        "text": "Roni",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102803",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102803",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102803"
      }
    ],
    "meta": {
      "lastUpdated": "2023-03-08T13:13:49.966892+00:00",
      "profile": [
        "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
      ],
      "versionId": "MTY3ODI4MTIyOTk2Njg5MjAwMA"
    },
    "name": [
      {
        "text": "Hulk",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102778",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102778",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102778"
      }
    ]
  }
}
```

```
],
"meta": {
  "lastUpdated": "2023-03-08T12:18:53.600353+00:00",
  "profile": [
    "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
  ],
  "versionId": "MTY3ODI3NzkzMzYwMDM1MzAwMA"
},
"name": [
  {
    "text": "Stark",
    "use": "official"
  }
],
"resourceType": "Patient"
}
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102767",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102767",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102767"
      }
    ],
    "meta": {
      "lastUpdated": "2023-03-08T11:11:12.338686+00:00",
      "profile": [
        "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
      ],
      "versionId": "MTY3ODI3Mzg3MjMzODY4NjAwMA"
    },
    "name": [
      {
        "text": "Toni",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102756",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102756",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102756"
      }
    ]
  }
}
```

```
],
"meta": {
  "lastUpdated": "2023-03-08T11:06:36.858381+00:00",
  "profile": [
    "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
  ],
  "versionId": "MTY3ODI3MzU5Njg1ODM4MTAwMA"
},
"name": [
  {
    "text": "Budi",
    "use": "official"
  }
],
"resourceType": "Patient"
}
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102745",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102745",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102745"
      }
    ],
    "meta": {
      "lastUpdated": "2023-03-08T11:05:37.879975+00:00",
      "profile": [
        "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
      ],
      "versionId": "MTY3ODI3MzUzNzg3OTk3NTAwMA"
    },
    "name": [
      {
        "text": "Mardigu",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102734",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102734",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102734"
      }
    ]
  }
}
```

```
    ],
    "meta": {
      "lastUpdated": "2023-03-08T11:00:54.720631+00:00",
      "profile": [
        "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
      ],
      "versionId": "MTY3ODI3MzI1NDcyMDYzMTAwMA"
    },
    "name": [
      {
        "text": "Dodii",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
},
{
  "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02029102723",
  "resource": {
    "active": true,
    "birthDate": "1920-02-28",
    "id": "P02029102723",
    "identifier": [
      {
        "system": "https://fhir.kemkes.go.id/id/nik-ibu",
        "use": "official",
        "value": "367400001111222"
      },
      {
        "system": "https://fhir.kemkes.go.id/id/ihs-number",
        "use": "official",
        "value": "P02029102723"
      }
    ],
    "meta": {
      "lastUpdated": "2023-03-08T10:59:48.626894+00:00",
      "profile": [
        "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
      ],
      "versionId": "MTY3ODI3MzE4ODYyNjg5NDAwMA"
    },
    "name": [
      {
        "text": "Anang",
        "use": "official"
      }
    ],
    "resourceType": "Patient"
  }
}
],
"link": [
  {
    "relation": "search",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik-ibu%7C367400001111222"
  },
  {
    "relation": "first",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik-ibu%7C367400001111222"
  },
  {
    "relation": "self",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?identifier=https%3A%2F%2Ffhir.kemkes.go.id%2Fid%2Fnik-ibu%7C367400001111222"
  }
]
```

```
],  
"resourceType": "Bundle",  
"total": 10,  
"type": "searchset"  
}
```

Contoh Data Pencarian Berdasarkan Nama, Tanggal Lahir, dan Gender Pasien

```
{  
  "entry": [  
    {  
      "fullUrl": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/P02478375538",  
      "resource": {  
        "active": true,  
        "address": [  
          {  
            "city": "MUNA",  
            "country": "ID",  
            "extension": [  
              {  
                "extension": [  
                  {  
                    "url": "province",  
                    "valueCode": "74"  
                  },  
                  {  
                    "url": "city",  
                    "valueCode": "3"  
                  },  
                  {  
                    "url": "district",  
                    "valueCode": "26"  
                  },  
                  {  
                    "url": "village",  
                    "valueCode": "1008"  
                  },  
                  {  
                    "url": "rw",  
                    "valueCode": "1"  
                  },  
                  {  
                    "url": "rt",  
                    "valueCode": "1"  
                  }  
                ]  
              }  
            ],  
            "url": "https://fhir.kemkes.go.id/r4/StructureDefinition/administrativeCode"  
          }  
        ],  
        "line": [  
          "KEL.TOMBU***"  
        ],  
        "use": "home"  
      }  
    },  
    {  
      "birthDate": "1980-12-03",  
      "communication": [  
        {  
          "language": {  
            "coding": [  
              {  
                "code": "id-ID",  
                "display": "Indonesian",  
                "system": "urn:ietf:bcp:47"  
              }  
            ]  
          }  
        }  
      ]  
    }  
  ]  
}
```

```
    ],
    "text": "Indonesian"
  },
  "preferred": true
},
],
"deceasedBoolean": false,
"extension": [
  {
    "url": "https://fhir.kemkes.go.id/r4/StructureDefinition/citizenshipStatus",
    "valueCode": "WNI"
  }
],
"gender": "male",
"id": "P02478375538",
"identifier": [
  {
    "system": "https://fhir.kemkes.go.id/id/ihs-number",
    "use": "official",
    "value": "P02478375538"
  },
  {
    "system": "https://fhir.kemkes.go.id/id/nik",
    "use": "official",
    "value": "#####"
  }
],
"link": [
  {
    "other": {
      "reference": "RelatedPerson/7af11905-8d49-47dc-a72b-bca4948c01e7"
    },
    "type": "refer"
  }
],
"meta": {
  "lastUpdated": "2023-05-31T06:40:40.038378+00:00",
  "profile": [
    "https://fhir.kemkes.go.id/r4/StructureDefinition/Patient"
  ],
  "versionId": "MTY4NTUxNTI0MDAzODM3ODAwMA"
},
"multipleBirthBoolean": false,
"name": [
  {
    "text": "patient 1",
    "use": "official"
  }
],
"resourceType": "Patient"
}
],
"link": [
  {
    "relation": "search",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?birthdate=1980-12-03&gender=male&name=patient+1"
  },
  {
    "relation": "first",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?birthdate=1980-12-03&gender=male&name=patient+1"
  },
  {
    "relation": "self",
    "url": "https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/?birthdate=1980-12-03&gender=male&name=patient+1"
  }
]
```

```
],
"resourceType": "Bundle",
"total": 1,
"type": "searchset"
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau meminta data dengan format, parameter, atau ketentuan lainnya yang tidak sesuai atau tidak dimengerti oleh sistem.

Contoh Data

```
{
  "resourceType": "OperationOutcome",
  //data.terkait.resource.OperationOutcome
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

1.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --insecure --location ^
--header "Authorization: Bearer lWnwCNpXQdGpyOTvJpyT1iTJeyp" ^
--request GET ^
"https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient?identifier=https://fhir.kemkes.go.id/id/nik|3171012345678901"
```

cURL (Linux)

```
curl --insecure --location \
--header 'Authorization: Bearer lWnwCNpXQdGpyOTvJpyT1iTJeyp' \
--request GET \
'https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient?identifier=https://fhir.kemkes.go.id/id/nik|3171012345678901'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient
```

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
 - c. pada pilihan **Type**, pilih **Bearer Token**,
 - d. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Params**, di bagian **Query Params**:
 - a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
 - d. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API ini,
 - e. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - f. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
6. Klik tombol **Send**.
7. Hasil *response* akan ditampilkan di bagian **Response**.

2. Patient - Detail Data

Fungsi dari ReST API ini adalah untuk mendapatkan data terkait *resource Patient* yang tersedia di *platform SATUSEHAT*. Untuk mendapatkan data yang dimaksud, nilai ID dari *resource Patient* tersebut **PERLU** diketahui dan disediakan sebagai parameternya.



Setiap terdapat simbol asterik * sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

2.1. Request

URL

```
https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient:id
```

HTTP Verb/Method

GET

Parameter Path URI

Nama Parameter	Tipe Data	Keterangan
*:id	uuid	ID referensi dari <i>resource Patient</i> yang akan dilihat detailnya.

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <code><access_token></code> . Nilai dari variabel <code><access_token></code> didapatkan dari properti <code>access_token</code> pada <i>object</i> dari hasil <i>response</i> JSON setelah proses autentikasi.
*Content-Type	string	<i>Mime type</i> dari <i>payload</i> data yang akan dikirimkan di dalam <i>body</i> dalam format JSON, WAJIB diisi dengan <code>application/json</code> .

2.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola `2xx` atau `4xx`, yang dikembalikan dari server mempunyai parameter *Content-Type* dengan nilai `application/json` di salah satu parameter *header*-nya.

2xx Success

Bila *resource Patient* dengan ID terkait berhasil ditemukan atau tersedia, maka akan mengembalikan data dari *resource Patient* yang tersimpan di *platform* SATUSEHAT.

Contoh Data

```
{
  "resourceType": "Patient",
  "id": "100000000001",
  //data.terkait.resource.Patient
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau meminta data dengan format, parameter, atau ketentuan lainnya yang tidak sesuai atau tidak dimengerti oleh sistem.

Contoh Data

```
{
  "resourceType": "OperationOutcome",
  //data.terkait.resource.OperationOutcome
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

2.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --insecure --location ^
--header "Authorization: Bearer lWnwCNpXQqdGpyOTvJpyT1iTJeyp" ^
--request GET ^
"https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/100000000001"
```


cURL (Linux)

```
curl --insecure --location \
--header 'Authorization: Bearer lWnwCNpXQqdGpyOTvJpyT1iTJeyp' \
--request GET \
'https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/100000000001'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient:id

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
 - c. pada pilihan **Type**, pilih **Bearer Token**,
 - d. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Params**, di bagian **Path Variables**:
 - a. Isi nilai parameter **id** dengan ID dari *resource Patient* yang ingin didapatkan datanya.
6. Klik tombol .
7. Hasil *response* akan ditampilkan di bagian **Response**.

3. Patient - Penambahan Data

Fungsi dari ReST API ini adalah untuk melakukan penambahan data terkait *resource Patient* ke dalam platform SATUSEHAT.



Setiap terdapat simbol asterik * sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: **variabel*.

3.1. Request

URL

`https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient:`

HTTP Verb/Method

POST

Header

Nama Parameter	Tipe Data	Keterangan
<i>*Authorization</i>	string	Header ini WAJIB diisi dengan nilai sesuai format: <i>Bearer <access_token></i> . Nilai dari variabel <i><access_token></i> didapatkan dari properti <i>access_token</i> pada <i>object</i> dari hasil <i>response</i> JSON setelah proses autentikasi.
<i>*Content-Type</i>	string	<i>Mime type</i> dari <i>payload</i> data yang akan dikirimkan di dalam <i>body</i> dalam format JSON, WAJIB diisi dengan <i>application/json</i> .

Body (*application/json*), dengan *identifier* NIK Pasien

Di bagian *body* ini *payload* JSON dari *resource Patient* sesuai standar FHIR dimasukkan. Terkait cara pengisian dari format FHIR tersebut di luar cakupan dari dokumentasi ini, silakan melihat pada dokumentasi terkait **Petunjuk Teknis** atau **Playbook** yang telah disediakan oleh tim SATUSEHAT dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Bentuk umum dari *payload* untuk penambahan data sebagai berikut:

```
{
  "resourceType": "Patient",
  "identifier": [
    {
      "use": "official",
      "system": "https://fhir.kemkes.go.id/id/nik",
      "value": "3171012345678901"
    },
    //data.terkait.identifier.lainnya
  ],
  //data.terkait.resource.Patient
}
```

Body (*application/json*), dengan *identifier* NIK Ibu

Di bagian *body* ini *payload* JSON dari *resource* *Patient* dengan *identifier* NIK Ibu sesuai standar FHIR dimasukkan. Terkait cara pengisian dari format FHIR tersebut di luar cakupan dari dokumentasi ini, silakan melihat pada dokumentasi terkait **Petunjuk Teknis** atau **Playbook** yang telah disediakan oleh tim SATUSEHAT dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Bentuk umum dari *payload* untuk penambahan data sebagai berikut:

```
{
  "resourceType": "Patient",
  "identifier": [
    {
      "use": "official",
      "system": "https://fhir.kemkes.go.id/id/nik-ibu",
      "value": "367400001111222"
    },
    //data.terkait.indentifier.lainnya
  ],
  //data.terkait.resource.Patient
}
```

3.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola *2xx* atau *4xx*, yang dikembalikan dari server mempunyai parameter *Content-Type* dengan nilai *application/json* di salah satu parameter *header*-nya.

2xx Success

Dari hasil *response* ini, **PERLU** disimpan nilai *UUID* yang didapat dari properti *id*, dimana nilai tersebut kemungkinan akan digunakan dalam proses lainnya yang terkait *resource Patient* ini.

Contoh Data

```
{
  "resourceType": "Patient",
  "id": "100000000001",
  //data.terkait.resource.Patient
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan lainnya yang tidak sesuai atau tidak dimengerti oleh sistem.

Contoh Data

```
{
  "resourceType": "OperationOutcome",
  //data.terkait.resource.OperationOutcome
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

3.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --insecure --location ^
--header "Authorization: Bearer lWnwCNpXQdGpyOTvJpyT1iTJeyP" ^
--header "Content-Type: application/json" ^
--data-raw "{
  \"resourceType\": \"Patient\",
  ...
}" ^
--request POST ^
"https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient"
```

cURL (Linux)

```
curl --insecure --location \
--header 'Authorization: Bearer lWnwCNpXQdGpyOTvJpyT1iTJeyP' \
--header 'Content-Type: application/json' \
--data-raw '{
  "resourceType": "Patient",
  ...
}' \
--request POST \
'https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient:

3. Lalu pilih *request method* **POST**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.

- c. pada pilihan **Type**, pilih **Bearer Token**,
- d. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Body**:
 - a. pilih **raw**,
 - b. kemudian disamping nilai tadi pilih **JSON**,
 - c. terakhir masukan *resource* JSON dari *Patient* yang akan diproses ke kotak masukan di bawah pilihan tadi. Contoh:


```
Unresolved directive in snippets/apidoc/file-example.adoc - include::C:/dto-technical-writer/dto-api-mpi/resources/api/products/add/req.jsonc[]
```
6. Klik tombol **Send**.
7. Hasil *response* akan ditampilkan di bagian **Response**.

4. Patient - Pembaruan Sebagian Data

Fungsi dari ReST API ini adalah untuk melakukan perubahan sebagian dari data terkait *resource Patient* ke dalam *platform* SATUSEHAT, yang sebelumnya sudah ditambahkan dan tersedia di dalam *platform* SATUSEHAT. Untuk melakukan perubahan sebagian (*patching*) data, **PERLU** ID dari *resource Patient* yang akan diubah dan juga nama/ID elemen dari *Patient* yang akan dilakukan perubahan.



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

4.1. Request

URL

```
https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/:id
```

HTTP Verb/Method

PATCH

Parameter Path URI

Nama Parameter	Tipe Data	Keterangan
*:id	uuid	ID referensi dari <i>resource Patient</i> yang akan dilakukan proses pembaruan data (<i>update</i>).

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <code><access_token></code> . Nilai dari variabel <code><access_token></code> didapatkan dari properti <code>access_token</code> pada <code>object</code> dari hasil <i>response</i> JSON setelah proses autentikasi.
*Content-Type	string	<i>Mime type</i> dari <i>payload</i> data yang akan dikirimkan di dalam <i>body</i> dalam format JSON, WAJIB diisi dengan <code>application/json</code> .

Body (`application/json`)

Di bagian *body* ini *payload* JSON dari *resource Patient* sesuai standar FHIR dimasukkan. Terkait cara pengisian dari format FHIR tersebut di luar cakupan dari dokumentasi ini, silakan melihat pada dokumentasi terkait **Petunjuk Teknis** atau **Playbook** yang telah disediakan oleh tim SATUSEHAT dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Bentuk umum dari *payload* untuk *patching* sebagai berikut:

```
[
  {
    "op": "<operasi>",
    "path": "<element-path>",
    "value": "<nilai-baru>"
  }
]
```

Dari bentuk umum tersebut, nilai:

- `<operasi>`, saat ini hanya tersedia operasi `replace` saja, yaitu untuk mengganti nilai properti/element dari *resource Patient* dengan ID terkait.
- `<element-path>`, nama properti/element dari *resource Patient* dengan ID terkait yang akan diganti nilainya, dengan format `/<path>/<to>/<element>`. Simbol `/` sebagai pemisah dari nama properti/element yang dimaksud.

Contoh, misalkan pada *resource Patient* ini ada properti/element dengan nama `language`, yang dalam bentuk path FHIR `Patient.language`, maka penulisan nilai `<element-path>` ini adalah `/language`.

- `<nilai-baru>`, diisi dengan nilai pengantinya.

Sehingga contoh *payload*-nya sebagai berikut:

```
[
  {
    "op": "replace",
    "path": "/language",
    "value": "id"
  }
]
```

4.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Bila proses pembaruan data berhasil maka akan mengembalikan *payload* dari *resource Patient* yang sebelumnya telah dikirim.

Contoh Data

```
{
  "resourceType": "Patient",
  "id": "100000000001",
  //data.terkait.resource.Patient
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format, parameter, atau ketentuan lainnya yang tidak sesuai atau tidak dimengerti oleh sistem.

Contoh Data

```
{
  "resourceType": "OperationOutcome",
  //data.terkait.resource.OperationOutcome
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

4.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --insecure --location ^
  --header "Authorization: Bearer lWnwCNpXQqdGpyOTvJpyT1iTJey" ^
  --header "Content-Type: application/json" ^
```



```
--data-raw "[
{
  \"op\" : \"<operasi>\",
  \"path\" : \"<element-path>\",
  \"value\" : \"<nilai-baru>\"
}
]" ^
--request PATCH ^
"https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/100000000001"
```

cURL (Linux)

```
curl --insecure --location \
--header 'Authorization: Bearer lWnwCNpXQd6pyOTvJpyT1iTJeyp' \
--header 'Content-Type: application/json' \
--data-raw '[
{
  "op": "<operasi>",
  "path": "<element-path>",
  "value": "<nilai-baru>"
}
]' \
--request PATCH \
'https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/100000000001'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-satusehat-dev.dto.kemkes.go.id/fhir-r4/v1/Patient/:id
```

3. Lalu pilih *request method* **PATCH**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
 - c. pada pilihan **Type**, pilih **Bearer Token**,
 - d. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Params**, di bagian **Path Variables**:
 - a. Isi nilai parameter **id** dengan ID dari *resource Patient* yang akan diperbarui (*update*).
6. Pada tab **Body**:
 - a. pilih **raw**,
 - b. kemudian disamping nilai tadi pilih **JSON**,
 - c. terakhir masukan *payload* JSON untuk melakukan perubahan ke kotak masukan di bawah pilihan tadi. Contoh:

```
[
{
```

```
[{"op": "replace",  
  "path": "/language",  
  "value": "id"  
}]
```

7. Klik tombol **Send**.
8. Hasil *response* akan ditampilkan di bagian **Response**.

DAFTAR ISTILAH

API

Singkatan dari *Application Programming Interface*, yaitu kumpulan definisi dan protokol terkait koneksi antara komputer atau aplikasi yang memungkinkan untuk melakukan pertukaran data.

JWT

Singkatan dari *JSON Web Token*, yaitu suatu standar untuk pertukaran data yang diperlukan dalam proses autentikasi.

MSI

Data Fasyankes atau Master Sarana Index (MSI), merupakan standar data 35 fasyankes di Indonesia. Disusun dari berbagai sumber, seperti Sistem Informasi Sumber Daya Manusia Kesehatan (SI-SDMK), Sistem Informasi Manajemen Data Kefarmasian (SIMADA), RS Online, dan lain sebagainya.

REFERENSI

- [restfulapi.net] "ReST API Tutorial". <https://restfulapi.net>.
- [wikipedia-rest] "Representational State Transfer". https://en.wikipedia.org/wiki/Representational_state_transfer.
- [wikipedia-http-method] "Hypertext Transfer Protocol, Request_methods". https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods
- [wikipedia-http-status-code] "List of HTTP status codes". https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- [wikipedia-iso8601] "ISO 8601". https://en.wikipedia.org/wiki/ISO_8601
- [wikipedia-api] "Application Programming Interface". <https://en.wikipedia.org/wiki/API>
- [wikipedia-whitespace] "Whitespace character". https://en.wikipedia.org/wiki/Whitespace_character
- [wikipedia-fhir] "Fast_Healthcare_Interoperability_Resources", <https://en.wikipedia.org/wiki/>
- [hl7-fhir] "Welcome to FHIR. FHIR is a standard for health care data exchange, published by HL7®". <https://www.hl7.org/fhir/index.html>
- [satusehat-fhir] "SATUSEHAT FHIR R4 Implementation Guide". <https://simplifier.net/guide/SATUSEHAT-FHIR-R4-Implementation-Guide/Home?version=current>
- [admonitions] "Admonitions". <https://docs.asciidoctor.org/asciidoc/latest/blocks/admonitions>
- [atc-code] "Daftar Kode ATC" https://id.wikipedia.org/wiki/Kategori:Kode_ATC
- [atc-level] "Daftar Level ATC" <https://www.who.int/tools/atc-ddd-toolkit/atc-classification>