

Singapay B2B - Ringkasan Fitur Lengkap & User Access

Version: 1.0

Last Updated: December 8, 2025

Status: Production Ready

Ringkasan Singkat

Singapay B2B Payment Gateway adalah solusi pembayaran komprehensif yang memungkinkan merchant dan user untuk: - Menerima pembayaran via Virtual Account (VA), QRIS, dan Payment Links - Melakukan pencairan dana ke berbagai bank - Tracking transaksi real-time - Mengelola multiple accounts dengan dashboard terintegrasi

Apa Yang Bisa Dilakukan User

1. PENERIMA PEMBAYARAN (Merchant/Customer)

A. Membuat Request Export PDF dengan Pembayaran User dapat membuat request untuk export dokumen (Business Plan, Financial Plan, Forecast) dan membayarnya.

Alur:

1. User POST ke /api/pdf-export-pro/create-request
↓
2. Backend: Buat PDFExportRequest + Generate VA
↓
3. Response: Nomor VA, bank, amount, instruksi pembayaran
↓
4. User: Transfer ke nomor VA
↓
5. Backend (Webhook): Terima konfirmasi pembayaran
↓
6. User: Download PDF

Request Code:

```
curl -X POST http://localhost:8000/api/pdf-export-pro/create-request \
-H "Authorization: Bearer TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "export_type": "business_plan",
  "business_id": 1,
```

```
        "package": "professional"
    }'
```

Response:

```
{
  "status": "success",
  "data": {
    "export_request_id": 1,
    "va_number": "1234567890123456",
    "bank": "BRI",
    "amount": 50000,
    "expires_at": "2024-01-18",
    "reference_id": 1,
    "instructions": "Transfer ke 1234567890123456 sebesar Rp 50.000"
  }
}
```

B. Pilih Metode Pembayaran Alternatif User bisa memilih antara Virtual Account, QRIS, atau Payment Link.

Metode QRIS:

```
curl -X POST http://localhost:8000/api/pdf-export-pro/payment-options \
-H "Authorization: Bearer TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "export_request_id": 1,
  "payment_method": "qrис"
}'
```

Response:

```
{
  "status": "success",
  "data": {
    "status": "success",
    "qr_data": "00020101021226570015...",
    "qr_url": "https://qrис.singapay.id/...",
    "amount": 50000,
    "expired_at": "2024-01-25"
  }
}
```

Metode Payment Link:

```
curl -X POST http://localhost:8000/api/pdf-export-pro/payment-options \
-H "Authorization: Bearer TOKEN" \
```

```
-H "Content-Type: application/json" \
-d '{
    "export_request_id": 1,
    "payment_method": "payment_link"
}'
```

C. Cek Status Pembayaran User bisa cek status pembayaran kapan saja.

Code:

```
curl -X GET http://localhost:8000/api/pdf-export-pro/payment-status/1 \
-H "Authorization: Bearer TOKEN"
```

Response:

```
{
    "status": "success",
    "data": {
        "export_request_id": 1,
        "payment_status": "completed",
        "amount": 50000,
        "va_number": "1234567890123456",
        "paid_at": "2024-01-16T10:30:00",
        "pdf_download_url": "https://yourdomain.com/api/pdf-export-pro/download/1"
    }
}
```

D. Download PDF (Setelah Pembayaran) User bisa download PDF setelah pembayaran sukses.

Code:

```
curl -X GET http://localhost:8000/api/pdf-export-pro/download/1 \
-H "Authorization: Bearer TOKEN"
```

Response:

```
{
    "status": "success",
    "data": {
        "filename": "Business_Plan_2024.pdf",
        "download_url": "https://yourdomain.com/pdf-export/download-file/1"
    }
}
```

E. Lihat Riwayat Export User bisa melihat semua export yang pernah dibuat.

Code:

```
curl -X GET http://localhost:8000/api/pdf-export-pro/history \
-H "Authorization: Bearer TOKEN"
```

Response:

```
{
  "status": "success",
  "data": {
    "data": [
      {
        "id": 1,
        "export_type": "business_plan",
        "package": "professional",
        "amount": 50000,
        "status": "completed",
        "paid_at": "2024-01-16",
        "created_at": "2024-01-15"
      }
    ],
    "pagination": {
      "total": 1,
      "per_page": 10,
      "current_page": 1
    }
  }
}
```

F. Lihat Harga & Paket User bisa melihat semua paket yang tersedia.

Code:

```
curl -X GET http://localhost:8000/api/pdf-export-pro/pricing
```

Response:

```
{
  "status": "success",
  "data": {
    "free": {
      "name": "Standard (Free)",
      "price": 0,
      "features": [
        "Basic PDF export",
        "Customizable templates"
      ]
    }
  }
}
```

```

        "Watermark included",
        "Standard formatting"
    ],
    "watermark": true
},
"professional": {
    "name": "Professional",
    "price": 50000,
    "features": [
        "Professional PDF export",
        "No watermark",
        "Premium formatting",
        "Advanced charts"
    ],
    "watermark": false
},
"business": {
    "name": "Business",
    "price": 100000,
    "features": [
        "All Professional features",
        "Priority support",
        "Multiple exports",
        "Team collaboration"
    ],
    "watermark": false
}
}
}
}

```

2. PENGELOLA AKUN (Admin/Merchant Backend)

A. Kelola Account/Sub-Account Admin bisa membuat dan mengelola multiple sub-accounts untuk customers.

Buat Account:

```

// Backend Code - Using SingapayPDFExportService
$service = new SingapayPDFExportService();
$response = Http::post($apiUrl . '/api/v1.0/accounts', [
    'name' => 'Customer Name',
    'email' => 'customer@example.com',
    'phone' => '08123456789'
]);

```

```

// Response
{
    "id": "account_12345",
    "name": "Customer Name",
    "email": "customer@example.com",
    "phone": "08123456789",
    "status": "active",
    "created_at": "2024-01-15"
}

List Accounts:

$response = Http::withToken($accessToken)
    ->get($apiUrl . '/api/v1.0/accounts');

// Response: Array of accounts with pagination

Update Account Status:

$response = Http::patch($apiUrl . '/api/v1.0/accounts/update-status/account_12345', [
    'status' => 'inactive'
]);

```

B. Kelola Virtual Account (VA) Admin bisa membuat dan mengelola Virtual Account untuk setiap transaksi.

Buat VA Baru:

```

$response = Http::post($apiUrl . '/api/v1.0/virtual-accounts/account_12345', [
    'bank_code' => 'BRI',           // BRI, BNI, DANAMON, MAYBANK
    'amount' => 50000,
    'name' => 'Invoice #123',
    'kind' => 'temporary',         // temporary atau permanent
    'expired_at' => 1760643599000, // Millisecond timestamp (untuk temporary)
    'max_usage' => 1,              // Bisa dipakai berapa kali
    'currency' => 'IDR'
]);

// Response
{
    "id": "va_98765",
    "virtual_account_number": "1234567890123456",
    "bank_code": "BRI",
    "amount": 50000,
    "status": "active",
    "kind": "temporary",
    "expired_at": "2024-02-15",
}

```

```
        "max_usage": 1,  
        "created_at": "2024-01-15"  
    }  
}
```

List VA dari Account:

```
$response = Http::get($apiUrl . '/api/v1.0/virtual-accounts/account_12345');  
  
// Response: Array of VA dengan pagination
```

Update VA:

```
$response = Http::put($apiUrl . '/api/v1.0/virtual-accounts/account_12345/va_98765', [  
    'amount' => 75000,  
    'name' => 'Updated Invoice',  
    'status' => 'active'  
]);
```

C. Monitor VA Transactions (Pembayaran) Admin bisa melihat semua pembayaran yang masuk ke VA.

List VA Transactions:

```
$response = Http::get($apiUrl . '/api/v1.0/va-transactions/account_12345');  
  
// Response  
{  
    "data": [  
        {  
            "id": "tx_12345",  
            "va_number": "1234567890123456",  
            "bank": {  
                "short_name": "BRI",  
                "number": "002",  
                "swift_code": "BRINIDJA"  
            },  
            "amount": 50000,  
            "status": "paid",  
            "post_timestamp": "2024-01-16 10:30:00",  
            "processed_timestamp": "2024-01-16 10:31:00",  
            "fees": {  
                "name": "VA BRI",  
                "amount": 1500  
            }  
        }  
    ],  
    "pagination": {
```

```

        "total": 10,
        "per_page": 10,
        "current_page": 1
    }
}

```

Detail Transaction:

```
$response = Http::get($apiUrl . '/api/v1.0/va-transactions/account_12345/tx_12345');
```

D. Buat & Kelola Payment Links Admin bisa membuat hosted payment pages untuk invoice.

Buat Payment Link:

```

$response = Http::post($apiUrl . '/api/v1.0/payment-link-manage/account_12345', [
    'reff_no' => 'PL20240115A001',
    'title' => 'Invoice Payment #INV-2024-001',
    'required_customer_detail' => true,
    'max_usage' => 5,
    'expired_at' => 1760643599000, // Millisecond timestamp
    'total_amount' => 55000,
    'items' => [
        [
            'name' => 'Business Plan Document',
            'quantity' => 1,
            'unit_price' => 50000
        ],
        [
            'name' => 'Processing Fee',
            'quantity' => 1,
            'unit_price' => 5000
        ]
    ],
    'whitelisted_payment_method' => ['VA_BRI', 'QRIS', 'INDOMARET']
]);
// Response
{
    "id": "pl_12345",
    "reff_no": "PL20240115A001",
    "title": "Invoice Payment",
    "total_amount": 55000,
    "status": "open",
    "link": "https://payment.singapay.id/pl/abc123def456",
    "max_usage": 5,
}
```

```
        "expired_at": "2024-02-15",
        "created_at": "2024-01-15"
    }
```

Update Payment Link:

```
$response = Http::put($apiUrl . '/api/v1.0/payment-link-manage/account_12345/pl_12345', [
    'status' => 'closed',
    'max_usage' => 10,
    'expired_at' => 1761801600000
]);
```

List Payment Links:

```
$response = Http::get($apiUrl . '/api/v1.0/payment-link-manage/account_12345');
```

E. Monitor Payment Link Transactions Admin bisa melihat siapa saja yang sudah bayar via payment link.

List Payment Link History:

```
$response = Http::get($apiUrl . '/api/v1.0/payment-link-histories/account_12345');

// Response
{
    "data": [
        {
            "id": "history_12345",
            "transaction_reff_no": "TRX20240116001",
            "payment_method": "qrис",
            "amount": 55000,
            "customer_name": "John Doe",
            "customer_email": "john@example.com",
            "payment_link_title": "Invoice Payment",
            "status": "success",
            "post_timestamp": "2024-01-16 10:00:00",
            "processed_timestamp": "2024-01-16 10:02:00"
        }
    ]
}
```

Detail History:

```
$response = Http::get($apiUrl . '/api/v1.0/payment-link-histories/account_12345/history_12345');
```

F. Generate QRIS untuk Pembayaran Admin bisa generate QRIS code untuk pembayaran dinamis.

Generate QRIS:

```
$response = Http::post($apiUrl . '/api/v1.0/qrис-dynamic/account_12345/generate-qr', [
    'amount' => 50000,
    'expired_at' => '2025-08-30 12:00:00', // Format: Y-m-d H:i:s
    'tip_indicator' => 'fixed_amount', // atau 'percentage'
    'tip_value' => 2000 // Jumlah atau persentase
]);

// Response
{
    "id": 2,
    "reff_no": "6601K5WZZ71ZQ3KQJZZRV811XXHE",
    "status": "open",
    "type": "mpm-dynamic",
    "amount": 50000,
    "tip_amount": 2000,
    "total_amount": 52000,
    "qr_data": "000201010212...",
    "expired_at": "2025-12-30 12:00:00",
    "created_at": "2025-09-24 11:33:38"
}
```

List QRIS:

```
$response = Http::get($apiUrl . '/api/v1.0/qrис-dynamic/account_12345');
```

G. Cek Saldo Akun Admin bisa melihat saldo tersedia di account.

Check Balance:

```
$response = Http::get($apiUrl . '/api/v1.0/balance-inquiry/account_12345');

// Response
{
    "status": 200,
    "success": true,
    "data": {
        "pending_balance": {
            "value": "0.00",
            "currency": "IDR"
        },
        "held_balance": {
            "value": "0.00",

```

```

        "currency": "IDR"
    },
    "available_balance": {
        "value": "500000.00",
        "currency": "IDR"
    },
    "balance": {
        "value": "500000.00",
        "currency": "IDR"
    }
}

```

3. PENCAIRAN DANA (Withdrawal/Disbursement)

A. Cek Biaya Transfer Admin bisa cek berapa biaya untuk transfer ke bank tertentu.

Check Fee:

```

$response = Http::post($apiUrl . '/api/v1.0/disbursement/account_12345/check-fee', [
    'amount' => 50000,
    'bank_swift_code' => 'BRINIDJA' // Kode SWIFT bank tujuan
]);

// Response
{
    "status": 200,
    "success": true,
    "data": {
        "gross_amount": "50000",
        "transfer_fee": "3000.00",
        "net_amount": "47000",
        "currency": "IDR",
        "beneficiary": {
            "full_name": "Bank Rakyat Indonesia",
            "short_name": "BRI"
        }
    }
}

```

B. Verifikasi Rekening Tujuan Admin bisa verify nama pemilik rekening sebelum transfer.

Check Beneficiary:

```
$response = Http::post($apiUrl . '/api/v1.0/disbursement/check-beneficiary', [
    'bank_account_number' => '521398319083210',
    'bank_swift_code' => 'BRINIDJA'
]);

// Response
{
    "status": "valid",
    "bank_name": "Bank BRI",
    "bank_number_code": "002",
    "bank_swift_code": "BRINIDJA",
    "bank_account_number": "521398319083210",
    "bank_account_name": "Darsirah Vero Sirait"
}
```

C. Transfer Dana ke Bank Admin bisa melakukan withdrawal/pencairan ke berbagai bank.

Transfer Funds:

```
$timestamp = now()->toIso8601String();
$body = [
    'reference_number' => '123456789123', // Unique reference
    'amount' => 50000,
    'bank_swift_code' => 'BRINIDJA',
    'bank_account_number' => '521398319083210',
    'notes' => 'Bayar komisi penjualan'
];

$response = Http::withHeaders([
    'X-PARTNER-ID' => env('SINGAPAY_PARTNER_ID'),
    'Authorization' => "Bearer {$accessToken}",
    'X-Signature' => generateAsymmetricSignature($body, $timestamp),
    'X-Timestamp' => $timestamp
])->post($apiUrl . '/api/v1.0/disbursements/account_12345/transfer', $body);

// Response
{
    "status": 200,
    "success": true,
    "data": {
        "transaction_id": "1512220251105174015668",
        "reference_number": "123456789123",
    }
}
```

```

    "status": "success",
    "bank": {
        "code": "BRI",
        "account_name": "Darsirah Vero Sirait",
        "account_number": "521398319083210"
    },
    "gross_amount": {
        "value": "50000",
        "currency": "IDR"
    },
    "fee": {
        "name": "Transfer Fee",
        "value": "3000",
        "currency": "IDR"
    },
    "net_amount": {
        "value": "47000",
        "currency": "IDR"
    }
}
}

```

D. List Disbursement History Admin bisa melihat riwayat semua pencairan.

List Disbursements:

```
$response = Http::get($apiUrl . '/api/v1.0/disbursement/account_12345');
```

// Response: Array of disbursements dengan pagination

Detail Disbursement:

```
$response = Http::get($apiUrl . '/api/v1.0/disbursement/account_12345/tx_12345');
```

4. LAPORAN & REKONSILIASI (Reporting)

A. Lihat Statement/Mutasi Rekening Admin bisa melihat semua pergerakan saldo di account.

List Statements:

```
$response = Http::get($apiUrl . '/api/v1.0/statements/account_12345');
```

// Response
{

```

"data": [
{
    "id": "stmt_12345",
    "transaction_id": "tx_12345",
    "type": "credit", // credit atau debit
    "kind": "settlement", // settlement, disbursement, fee, etc
    "amount": {
        "value": "50000",
        "currency": "IDR"
    },
    "balance_after": {
        "value": "500000",
        "currency": "IDR"
    },
    "processed_timestamp": "2024-01-16 10:30:00",
    "notes": "VA Settlement - INV-123"
}
]
}

```

B. Export Laporan Admin bisa export laporan untuk rekonsiliasi dengan accounting system.

Statement Detail:

```
$response = Http::get($apiUrl . '/api/v1.0/statements/account_12345/stmt_12345');
```

5. NOTIFIKASI & WEBHOOK

A. Terima Notifikasi Pembayaran (Real-time) Backend otomatis receive webhook ketika pembayaran masuk.

Webhook Handler (Laravel):

```
// app/Http/Controllers/Singapay/WebhookController.php

public function handlePaymentSettlement(Request $request)
{
    // Validate signature
    $signature = $request->header('X-Singapay-Signature');
    $body = $request->getContent();
    $expectedSignature = hash_hmac('sha256', $body, config('services.singapay.webhook_secret'));

    if (!hash_equals($expectedSignature, $signature)) {
        return response()->json(['error' => 'Invalid signature'], 401);
    }
}
```

```

}

$data = $request->json('data');

// Update export request status
$exportRequest = PDFExportRequest::where('singapay_reference_id', $data['transaction'][
    ->first());

if ($exportRequest) {
    $exportRequest->update([
        'status' => 'completed',
        'paid_at' => now()
    ]);
}

// Send confirmation email
Mail::send(new PaymentConfirmationMail($exportRequest));

Log::info('Payment received', [
    'export_id' => $exportRequest->id,
    'amount' => $data['transaction']['amount']['value']
]);
}

return response()->json(['success' => true], 200);
}

```

Webhook Payload (VA Payment):

```
{
    "status": 200,
    "success": true,
    "data": {
        "transaction": {
            "reff_no": "3211120250926133543246",
            "type": "va",
            "status": "paid",
            "amount": {
                "value": "100000.00",
                "currency": "IDR"
            },
            "post_timestamp": "26 Sep 2025 13:35:45"
        },
        "customer": {
            "id": "cust_123",
            "name": "John Doe",
            "email": "john@example.com",
            "phone": "08123456789"
        }
    }
}
```

```

        },
        "payment": {
            "method": "va",
            "additional_info": {
                "va_number": "7872955146576837",
                "bank": {
                    "short_name": "BRI",
                    "swift_code": "BRINIDJA"
                }
            }
        }
    }
}

```

6. CARDLESS WITHDRAWAL (ATM Tanpa Kartu)

Admin bisa membuat withdrawal untuk customer via ATM tanpa kartu.

Create CLWD:

```
$response = Http::post($apiUrl . '/api/v1.0/cardless-withdrawals/account_12345', [
    'withdraw_amount' => 1000000,
    'payment_vendor_code' => 'CLWD_ALTO' // Vendor khusus CLWD
]);
```

List CLWD:

```
$response = Http::get($apiUrl . '/api/v1.0/cardless-withdrawals/account_12345');
```

Cancel CLWD:

```
$response = Http::patch($apiUrl . '/api/v1.0/cardless-withdrawals/account_12345/cancel/clwd_
```

Security & Validation

1. Signature Validation (Webhook)

```
// HMAC SHA256 untuk VA/QRIS
$clientId = env('SINGAPAY_CLIENT_ID');
$body = $request->getContent();
$sortedBody = json_encode(json_decode($body, true), JSON_SORT_KEYS);
$expectedSignature = hash_hmac('sha256', $sortedBody, $clientId);
hash_equals($expectedSignature, $request->header('X-Signature'));
```

2. Asymmetric Signature (Disbursement)

```
// HMAC SHA512 dengan timestamp
$timestamp = now()->toIso8601String();
$method = 'POST';
$endpoint = '/api/v1.0/disbursements/account_12345/transfer';
$sha256Hash = hash('sha256', json_encode($data));

$signatureData = strtoupper($method) . ':' . $endpoint . ':' .
    $accessToken . ':' . $sha256Hash . ':' . $timestamp;

$signature = base64_encode(
    hash_hmac('sha512', $signatureData, env('SINGAPAY_CLIENT_SECRET'), true)
);
```

Supported Banks

Virtual Account

- **BRI** - Bank Rakyat Indonesia (BRINIDJA)
- **BNI** - Bank Negara Indonesia (BNINIDJA)
- **DANAMON** - Bank Danamon (BDINIDJA)
- **MAYBANK** - Maybank Indonesia (IBBKIDJA)

Disbursement (60+ banks)

BCA, Mandiri, BTN, Permata, CIMB Niaga, DBS, Panin, Syariah Banks, BEI, OCBC, HSBC, Citibank, Standard Chartered, dan banyak lagi.

Implementation Checklist

- Get Singapay credentials (Partner ID, Client ID, Secret)
- Setup environment variables
- Create models: PDFExportRequest, SingapayPayment
- Create controllers: PDFExportProController, WebhookController
- Create service: SingapayPDFExportService
- Setup database migrations
- Register routes in routes/api.php
- Register webhook in Singapay dashboard
- Test create export request
- Test payment flow (VA/QRIS/Link)
- Test webhook signature validation
- Test payment status check
- Test PDF download

- Test disbursement (fee check, transfer)
 - Setup monitoring & logging
 - Deploy to production
-

Next Steps

1. Environment Setup

```
SINGAPAY_PARTNER_ID=your_partner_id
SINGAPAY_CLIENT_ID=your_client_id
SINGAPAY_CLIENT_SECRET=your_secret
SINGAPAY_WEBHOOK_SECRET=your_webhook_secret
SINGAPAY_MERCHANT_ACCOUNT_ID=your_account_id
SINGAPAY_ENV=sandbox # atau production
```

2. Database Migration

```
php artisan migrate
```

3. Routes Configuration (Already done in routes/api.php)

```
Route::prefix('pdf-export-pro')->middleware(['auth:sanctum'])->group(function () {
    Route::post('/create-request', [PDFExportProController::class, 'createExportRequest'])
    Route::post('/payment-options', [PDFExportProController::class, 'generatePaymentOpt']
    Route::get('/payment-status/{export_request_id}', [PDFExportProController::class, 'getPaymentStatus'])
    Route::get('/download/{export_request_id}', [PDFExportProController::class, 'download'])
    Route::get('/history', [PDFExportProController::class, 'getExportHistory']);
    Route::get('/pricing', [PDFExportProController::class, 'getPackagePricing']);
});

Route::prefix('webhooks/singapay')->group(function () {
    Route::post('/payment-settlement', [WebhookController::class, 'handlePaymentSettlement'])
        ->withoutMiddleware('verify_csrf_token');
});
```

4. Testing

```
# Test create request
curl -X POST http://localhost:8000/api/pdf-export-pro/create-request \
-H "Authorization: Bearer TOKEN" \
-H "Content-Type: application/json" \
-d '{"export_type": "business_plan", "business_id": 1, "package": "professional"}'

# Test pricing
curl -X GET http://localhost:8000/api/pdf-export-pro/pricing
```

Documentation Files

- **SINGAPAY_B2B_PAYMENT_GATEWAY_GUIDE.md** - Complete API reference
 - **SINGAPAY_PDF_EXPORT_QUICK_START.md** - Quick setup guide
 - **SINGAPAY_PDF_EXPORT_PRO_ROUTES.md** - Route documentation
 - **SINGAPAY_FITUR LENGKAP.md** - This file (Feature overview)
-

Support

- **Singapay Docs:** <https://docs-payment-b2b.singapay.id>
 - **Sandbox API:** <https://sandbox-payment-b2b.singapay.id>
 - **Production API:** <https://payment-b2b.singapay.id>
 - **Contact:** [Singapay Support Team]
-